公理化集合论机器证明系统

数学机械化丛书

13

郁文生　孙天宇　付尧顺　著

科学出版社

数学机械化丛书 13

公理化集合论机器证明系统

郁文生 孙天宇 付尧顺 著

科 学 出 版 社

北 京

内 容 简 介

布尔巴基学派的序、代数、拓扑三大母结构是现代数学的基础. 利用计算机证明辅助工具, 可以完整构建这三大母结构的形式化系统. 本书利用交互式定理证明工具 Coq, 实现 Morse-Kelley 公理化集合论形式化系统, 包括对该体系中 8 个公理 (含选择公理)和 1 个公理图示以及全部 181 条定义或定理的Coq 描述, 其中构造了序数和基数, 定义了非负整数, 把Peano公设当作定理, 可以迅速而自然地给出一个数学基础, 摆脱了明显的悖论. 这是Morse-Kelley公理化集合论系统的首次形式化实现. 在Morse-Kelley公理化集合论形式化系统下, 作为应用, 我们给出选择公理与它的几个著名等价命题间等价性的机器证明, 这些命题包括 Tukey 引理、Hausdorff 极大原则、极大原则、Zorn 引理、良序定理及 Zermelo 假定等. 在我们开发的系统中, 全部定理无例外地给出 Coq 的机器证明代码, 所有形式化过程已被 Coq 验证, 并在计算机上运行通过, 体现了基于 Coq 的数学定理机器证明具有可读性和交互性的特点, 其证明过程规范、严谨、可靠. 该系统可方便地应用于拓扑学和代数学理论的形式化构建.

本书可作为数学与计算机科学、信息科学相关专业的高年级本科生或研究生教材, 也可供从事人工智能相关科研工作者使用.

图书在版编目(CIP)数据

公理化集合论机器证明系统/郁文生, 孙天宇, 付尧顺著. —北京: 科学出版社, 2020.3

(数学机械化丛书; 13)

ISBN 978-7-03-064039-0

Ⅰ. ①公… Ⅱ. ①郁… ②孙… ③付… Ⅲ.①集论公理系统–机器证明 Ⅳ. ①O144

中国版本图书馆 CIP 数据核字 (2019) 第 295847 号

责任编辑: 王丽平 / 责任校对: 邹慧卿
责任印制: 吴兆东 / 封面设计: 陈 敬

科学出版社 出版
北京东黄城根北街 16 号
邮政编码: 100717
http://www.sciencep.com

北京源海印刷有限责任公司 印刷
科学出版社发行 各地新华书店经销
*
2020 年 3 月第 一 版 开本: 720 × 1000 1/16
2020 年12月第三次印刷 印张: 19 3/4 插页: 1
字数: 300 000

定价: 128.00 元

(如有印装质量问题, 我社负责调换)

“数学机械化丛书”前言①

十六七世纪以来，人类历史上经历了一场史无前例的技术革命，出现了各种类型的机器，取代各种形式的体力劳动，使人类进入一个新时代. 几百年后的今天，电子计算机已可开始有条件地代替一部分特定的脑力劳动，因而人类已面临另一场更宏伟的技术革命，处在又一个新时代的前夕. 数学是一种典型的脑力劳动，它在这一场新的技术革命中，无疑将扮演一个重要的角色. 为了了解数学在当前这场革命中所扮演的角色，就应对机器的作用，以及作为数学的脑力劳动的方式，进行一定的分析.

1. 什么是数学的机械化

不论是机器代替体力劳动，或是计算机代替某种脑力劳动，其所以成为可能，关键在于所需代替的劳动已经“机械化”，也就是说已实现了刻板化或规格化. 正因为割麦、刈草、纺纱、织布的动作已经是机械化刻板化了的，因而可据此造出割麦机、刈草机、纺纱机、织布机来. 也正因为加减乘除开方等运算这一类脑力劳动，几千年来就已经是机械地刻板地进行的，才有可能使得 17 世纪的法国数学家 Pascal，利用齿轮传动造出了第一台机械计算机——加法机，并由 Leibniz 改进成为也能进行乘法的机器. 数学问题的机械化，就要求在运算或证明过程中，每前进一步之后，都有一个确定的、必须选择的下一步，这样沿着一条有规律的、刻板的道路，一直达到结论.

在中小学数学的范围里，就有着不少已经机械化了的课题. 除了四则、开方等运算外，解线性联立方程组就是一个很好的例子. 在中学用的数学课本中，往往介绍解线性方程组的各种“消去法”，其求解过程是一个按一定程序进行的计算过程，也就是一种机械的、刻板的过程. 根据这一过程编成程序，由电子计算机付诸实施，就可以不仅机器化而且达到自动化，在几分钟甚至几秒钟之内求出一个未知数多至上百个的线性方程组的解答来，这在手工计算几乎是不可能的. 如果用手工计算，即使是解只有三四个未知数的方程组，也将是繁琐而令人厌烦的. 现代化的国

---

① 20 世纪七八十年代之交，我尝试用计算机证明几何定理取得成功，由此提出了数学机械化的设想. 先后在一些通俗报告与写作中，解释数学机械化的意义与前景，例如 1978 年发表于《自然辩证法通讯》的“数学机械化问题”以及 1980 年发表于《百科知识》的“数学的机械化”. 二文都重载于 1995 年由山东教育出版社出版的《吴文俊论数学机械化》一书. 经过 20 多年众多学者的努力，数学机械化在各个方面都取得了丰富多彩的成就，并已出版了多种专著，汇集成现在的数学机械化丛书. 现据 1980 年的《百科知识》的“数学的机械化”一文，稍加修改并作增补，以代丛书前言.

防、经济建设中, 大量出现的例如网络一类的问题, 往往可归结为求解很多未知数的线性方程组. 这使得已经机械化了的线性方程解法在四个现代化中起着一种重要作用.

即使是不专门研究数学的人们, 也大都知道, 数学的脑力劳动有两种主要形式: 数值计算与定理证明 (或许还应包括公式推导, 但这终究是次要的). 著名的数理逻辑学家美国洛克菲勒大学教授王浩先生在一篇有名的《向机械化数学前进》的文章中, 曾列举了这两种数学脑力劳动的若干不同之点. 我们可以简略而概括地把它们对比一下:

计算	证明
易	难
繁	简
刻板	灵活
枯燥	美妙

计算, 如已经提到过的加、减、乘、除、开方与解线性方程组, 其所以虽繁而易, 根本原因正在于它已经机械化. 而证明的巧而难, 是大家都深有体会的, 其根本原因也正在于它并没有机械化. 例如, 我们在中学初等几何定理的证明中, 就经常要依靠诸如直观、洞察、经验以及其他一些模糊不清的原则, 去寻找捷径.

2. 从证明的机械化到机器证明

一个值得提出的问题是: 定理的证明是不是也能像计算那样机械化, 因而把巧而难的证明, 化为计算那样虽繁而易的劳动呢? 事实上, 这一证明机械化的设想, 并不始自今日, 它早就为 17 世纪时的大哲学家、大思想家和大数学家 Descartes 和 Leibniz 所具有. 只是直到 19 世纪末, Hilbert(德国数学家, 1862～1943) 等创立并发展了数理逻辑以来, 这一设想才有了明确的数学形式. 又由于 20 世纪 40 年代电子计算机的出现, 才使这一设想的实现有了现实可能性.

从 20 世纪二三十年代以来, 数理逻辑学家们对于定理证明机械化的可能性进行了大量的理论探讨, 他们的结果大都是否定的. 例如 Gödel 等的一条著名定理就说, 即使看来最简单的初等数论这一范围, 它的定理证明的机械化也是不可能的. 另一方面, 1950 年波兰数学家 Tarski 则证明了初等几何 (以及初等代数) 这一范围的定理证明, 却是可以机械化的. 只是 Tarski 的结果近于例外, 在初等几何及初等代数以外的大量结果都是反面的, 即机械化是不可能的. 1956 年以来美国开始了利用电子计算机做证明定理的尝试. 1959 年王浩先生设计了一个机械化方法, 用计算机证明了 Russell 等著的《数学原理》这一经典著作中的几百条定理, 只用了 9 分钟, 在数学与数理逻辑学界引起了轰动. 一时间, 机器证明的前景似乎

非常乐观. 例如 1958 年时就有人曾经预测：在 10 年之内计算机将发现并证明一个重要的数学新定理. 还有人认为, 如果这样, 则不仅许多著名哲学家与数学家如 Peano、Whitehead、Russell、Hilbert 以及 Turing 等人的梦想得以实现, 而且计算将成为科学的皇后, 人类的主人!

然而, 事情的发展却并不如预期那样美好. 尽管在 1976 年, 美国的 Haken 等人, 在高速计算机上用了 1200 小时的计算时间, 解决了数学家们 100 多年来所未能解决的一个著名难题 —— 四色问题, 因此而轰动一时, 但是, 这只能说明计算机作为定理证明的辅助工具有着巨大潜力, 还不能认为这样的证明就是一种真正的机器证明. 用王浩先生的说法, Haken 等关于四色定理的证明是一种使用计算机的特例机证, 它只适用于四色这一特殊的定理, 这与所谓基础机器证明之能适用于一类定理者有别. 后者才真正体现了机械化定理证明, 进而实现机器证明的实质. 另一方面, 在真正的机械化证明方面, 虽然 Tarski 在理论上早已证明了初等几何的定理证明是能机械化的, 还提出了据以造判定机也即是证明机的设想, 但实际上他的机械化方法非常繁, 繁到不可收拾, 因而远远不是切实可行的. 1976 年时, 美国做了许多在计算机上证明定理的实验, 在 Tarski 的初等几何范围内, 用计算机所能证明的只是一些近于同义反复的 “儿戏式” 的 “定理”. 因此, 有些专家曾经发出过这样悲观的论调: 如果专依靠机器, 则再过 100 年也未必能证明出多少有意义的新定理来.

3. 一条切实可行的道路

1976 年冬, 我们开始了定理证明机械化的研究. 1977 年春取得了初步成果, 证明初等几何主要一类定理的证明可以机械化. 在理论上说来, 我们的结果已包括在 Tarski 的定理之中. 但与 Tarski 的结果不同, 我们的机械化方法是切实可行的, 即使用手算, 依据机械化的方法逐步进行, 虽然繁复, 也可以证明一些艰深的定理.

我们的方法主要分两步, 第一步是引进坐标, 然后把需证定理中的假设与终结部分都用坐标间的代数关系来表示. 我们所考虑的定理局限于这些代数关系都是多项式等式关系的范围, 例如平行、垂直、相交、距离等关系都是如此. 这一步可以叫做几何的代数化. 第二步是通过代表假设的多项式关系把终结多项式中的坐标逐个消去, 如果消去的结果为零, 即表明定理正确, 否则再作进一步检查. 这一步完全是代数的, 即用多项式的消元法来验证.

上述两步都可以机械与刻板地进行. 根据我们的机械化方法编成程序, 以在计算机上实现机器证明, 并无实质上的困难. 事实上数学所某些同志以及国外的王浩先生都曾在计算机上试行过. 我们自己也曾在国产的长城 203 台式机上证明了像 Simson 线那样不算简单的定理. 1978 年初我们又证明了初等微分几何中主要的一类定理证明也可以机械化. 而且这种机械化方法也是切实可行的, 并据此用手算证

明了不算简单的一些定理.

从我们的工作中可以看出, 定理的机械化证明, 往往极度繁复, 与通常既简且妙的证明形成对照, 这种以量的复杂来换取质的困难, 正是利用计算机所需要的.

在电子计算机如此发展的今天, 把我们的机械化方法在计算机上实现不仅不难, 而且有一台微型的台式机也就够了. 就像我们曾经使用过的长城 203, 它的存数最多只能到 234 个 10 进位的 12 位数, 就已能用以证明 Simson 线那样的定理. 随着超大规模集成电路与其他技术的出现与改进, 微型机将愈来愈小型化而内存却愈来愈大, 功能愈来愈多, 自动化的程度也愈来愈高. 进入 21 世纪以后, 这一类方便的小型机器将为广大群众普遍使用. 它们不仅将成为证明一些不很简单的定理的武器, 而且还可用以发现并证明一些艰深的定理, 而这种定理的发现与证明, 在数学研究手工业式的过去, 将是不可想像的. 这里我们应该着重指出, 我们并不鼓励以后人们将使用计算机来证明甚至发现一些有趣的几何定理. 恰恰相反, 我们希望人们不再从事这种虽然有趣却即是对数学甚至几何学本身也已意义不大的工作, 而把自己从这种工作中解放出来, 把自己的聪明才智与创造能力贯注到更有意义的脑力劳动上去.

还应该指出, 目前我们所能证明的定理, 局限于已经发现的机械化方法的范围, 例如初等几何与初等微分几何之内. 而如何超出与扩大这些机械化的范围, 则是今后需要探索的长期的理论性工作.

4. 历史的启示与中国古代数学

我们发现几何定理证明的机械化方法是在 1976 年至 1977 年之间. 约在两年之后我们发现早在 1899 年出版的 Hilbert 的经典名著《几何基础》中, 就有着一条真正的正面的机械化定理：初等几何中只涉及从属与平行关系的定理证明可以机械化. 当然, 原来的叙述并不是以机械化的语言来表达的, 也许就连 Hilbert 本人也并没有对这一定理的机械化意义有明确的认识, 自然更不见得有其他人提到过这一定理的机械化内容. Hilbert 是以公理化的典范而著称于世的, 但我认为, 该书更重要处, 是在于提供了一条从公理化出发, 通过代数化以到达机械化的道路. 自然, 处于 Hilbert 以及其后数学的一张纸一支笔的手工作业时代里, 公理化的思想与方法得到足够的重视与充分的发展, 而机械化的方向与意义受到数学家的忽视是完全可以理解的. 但电子计算机已日益普及, 因而繁琐而重复的计算已成为不足道的事情, 机械化的思想应比公理化思想受到更大重视, 似乎是合乎实际的.

其次应该着重指出, 我们从事机械化定理证明工作获得成果之前, 对 Tarski 的已有工作并无接触, 更没有想到 Hilbert 的《几何基础》会与机械化有任何关系. 我们是在中国古代数学的启发之下提出问题并想出解决办法来的.

说起来道理也很简单：中国的古代数学基本上是一种机械化的数学. 四则运算

与开方的机械化算法由来已久. 汉初完成的《九章算术》中, 对开平、立方与解线性联立方程组的机械化过程, 都有详细说明. 宋代更发展到高次代数方程求数值解的机械化算法.

总之, 各个数学领域都有定理证明的问题, 并不限于初等几何或微分几何. 这种定理证明肇始于古希腊的 Euclid 传统, 现已成为近代纯粹数学或核心数学的主流. 与之相异, 中国的古代学者重视的是各种问题特别是来自实际要求的具体问题的解决. 各种问题的已知数据与要求的数据之间, 很自然地往往以多项式方程的形式出现. 因之, 多项式方程的求解问题, 也就自然成为中国古代数学家研究的中心问题. 从秦汉以来, 所研究的方程由简到繁, 不断有所前进, 有所创新. 到宋元时期, 更出现了一个思想与方法的飞跃：天元术的创立.

“天元术” 到元代朱世杰时又发展成四元术, 所引入的天元、地元、人元、物元实际上相当于近代的未知元或未知数. 将这些未知元作为通常的已知数那样加减乘除, 就可得到与近代多项式和有理函数相当的概念与相应的表达形式和运算法则. 一些几何性质与关系很容易转化成这种多项式或有理函数的形式及其关系. 这使得过去依题意列方程这种无法可循需要高度技巧的工作从此变成轻而易举. 朱世杰 1303 年的《四元玉鉴》又给出了解任意多至四个未知元的多项式方程组的方法. 这里限于 4 个未知元只是由于所使用的计算工具 (算筹和算板) 的限制. 实质上他解方程的思想路线与方法完全可以适用于任意多的未知元.

不问可知, 在当时的具体条件下, 朱世杰的方法有许多缺陷. 首先, 当时还没有复数的概念, 因之朱世杰往往限于求出 (正) 实值. 这无可厚非, 甚至在 17 世纪 Descartes 的时代也还往往如此. 但此外朱世杰在方法上也未臻完善. 尽管如此, 朱世杰的思想路线与方法步骤是完全正确的, 我们在 20 世纪 70 年代之末, 遵循朱世杰的思想与方法的基本实质, 采用美国数学家 Ritt 在 1932 年、1950 年关于微分方程代数研究书中所提供的某些技术, 得出了解任意复多项式方程组的一般算法, 并给出了全部复数解的具体表达形式. 此后又得出了实系数时求实解的方法, 为重要的优化问题提供了一个具体的方法.

由于多种问题往往自然导致多项式方程组的求解, 因而我们解方程的一般方法可被应用于形形式式的问题. 这些问题可以来自数学自身, 也可以来自其他自然科学或工程技术. 在本丛书的第一本书, 吴文俊的《数学机械化》一书中, 可以看到这些应用的实例. 在工程技术方面的应用, 在本丛书中已有高小山的《几何自动作图与智能 CAD》与陈发来和冯玉瑜的《代数曲面造型》两本专著. 上述解多项式方程组的一般方法已推广至微分方程的情形. 许多应用以及相应论著正在酝酿之中.

5. 未来的技术革命与时代的使命

宋元时代天元术与四元术的创造, 把许多问题特别是几何问题转化成代数方程

与方程组的求解问题. 这一方法用于几何可称为几何的代数化. 12 世纪的刘益将新法与 “古法” 比较, 称 “省功数倍”，这可以说是减轻脑力劳动使数学走上机械化的道路的一项伟大的成就.

与天元术的创造相伴, 宋元时代的数学又引进了相当于现代多项式的概念, 建立了多项式的运算法则和消元法的有关代数工具, 使几何代数化的方法得到了有系统的发展, 见于宋元时代幸以保存至今的杨辉、李冶、朱世杰的许多著作之中. 几何的代数化是解析几何的前身, 这些创造使我国古代数学达到了又一个高峰. 可以说, 当时我国已到达了解析几何与微积分的大门, 具备了创立这些数学关键领域的条件, 但是各种原因使我们数学的雄伟步伐就在这些大门之前停顿下来. 几百年的停顿, 使我们这个古代的数学大国在近代变成了数学上的纯粹入超国家. 然而, 我国古代机械化与代数化的光辉思想和伟大成就是无法磨灭的. 本人关于数学机械化的研究工作, 就是在这些思想与成就启发之下的产物, 它是我国自《九章算术》以迄宋元时期数学的直接继承.

恩格斯曾经指出, 枪炮的出现消除了体力上的差别, 使中世纪的骑士阶级从此销声匿迹, 为欧洲从封建时代进入到资本主义时代准备了条件. 近年有些计算机科学家指出, 个人用计算机的出现, 其冲击作用可与枪炮的出现相比. 枪炮使人们在体力上难分强弱, 而个人用计算机将使人们在智力上难分聪明愚鲁. 又有人对数学的未来提出看法, 认为计算机的出现, 将使数学现在一张纸一支笔的方法, 在历史的长河中, 无异于石器时代的手工方法. 今天的数学家们, 不得不面对计算机的挑战, 但是, 也不必妄自菲薄. 大量繁复的事情交给计算机去做了, 人脑将仍然从事富有创造性的劳动.

我国在体力劳动的机械化革命中曾经掉队, 以致造成现在的落后状态. 在当前新的一场脑力劳动的机械化革命中, 我们不能重蹈覆辙. 数学是一种典型的脑力劳动, 它的机械化有着许多其他类型脑力劳动所不及的有利条件. 它的发扬与实现对我国的数学家是一种时代的使命. 我国古代数学的光辉, 鼓舞着我们为实现数学的机械化, 在某种意义上也可以说是真正的现代化而勇往直前.

吴文俊

2002 年 6 月于北京

前 言

公理化集合论是现代数学的基础, 集合论里普遍采用的公理体系是 ZFC 系统. 它由 Zermelo-Fraenkel (ZF) 集合论公理加上选择公理 (Axiom of Choice, AC) 构成, ZF 系统回避 "真类" 的对象, 成功地避免了 Russell 悖论. 然而, 美籍匈牙利数学家 von Neumann 认为, 集合悖论的出现不在于处理真类, 所有研究对象都是类. 类又分为两种: 集合与真类, 集合是可以成为其他类的元素的类, 真类是不可以成为任何类的元素的类. 德国学者 Bernays 于 1937 年以后遵循这一思想, 用 "类" 作为基本概念, 发展了公理集合论系统, 美籍奥地利数学家 Gödel 于 1939 年对之进行改进, 1940 年又利用这一系统证明了选择公理和连续统假设的相对相容性. 这一重要的公理集合论系统被称为 NBG 系统. NBG 系统是 ZF 系统的保守扩展.

1955 年, Kelley 在总结众多数学家 (Skolem, Morse, Hilbert, Bernays, von Neumann, Gödel, Tarski, Quine, Wang 等) 的思想基础上, 发展了被称为 Morse-Kelley (MK) 的公理集合论系统. 该系统承认存在比集合更广的类, 并采用无限的公理体系, 吸收了 ZFC 系统和 NBG 系统的优点, 可 "用来迅速而又自然地给出一个数学基础, 其中摆脱了明显的悖论". MK 系统是 ZFC 系统的一个真扩展. Mendelson, Monk 及 Rubin 等数学家均认为, MK 系统较 ZFC 系统和 NBG 系统应用起来更为便利.

关于形式化数学, 在 20 世纪初对数学基础的深入讨论中受到重视, 后来法国著名的布尔巴基学派对振兴法国当代数学起到了积极的推动作用, 也对整个 20 世纪数学的发展产生了深远的影响. 布尔巴基是一群以法国人为主的数学家的共同名字. 该学派提出数学结构的概念, 并用此概念统一现代数学. 按照布尔巴基学派的观点, 数学中有三大母结构, 即序结构、代数结构和拓扑结构. 基于这三大结构相互交融形成了现代数学的主体内容. 利用计算机证明辅助工具, 可以完整构建这三大母结构的形式化系统.

数学定理的计算机形式化证明, 近年来随着计算机科学的迅猛发展, 特别是证明辅助工具 Coq 的出现, 取得了长足的进展. Coq 的基本原理是归纳构造演算, 是一个交互式定理证明与程序开发系统, 可用于描述定理内容、验证定理证明. Coq 的交互式编译环境, 使用户以人机对话的方式一问一答, 可以边设计、边修改, 使证明中的错误及时得到补证. 进入 21 世纪以来, 随着四色定理、有限单群分类定理及 Kepler 猜想等一系列著名数学难题形式化证明的实现, 证明辅助工具 Coq 在学术界得到广泛认可. 2002 年菲尔兹奖获得者 Voevodsky 和 2010 年菲尔兹奖获得者

Villani 都大力倡导发展可信数学.

应当指出, 在数学定理机器证明方面, 由我国吴文俊院士开创的 “吴方法” 的研究在国际上是独具特色和领先的. 吴文俊院士早年留学法国, 深受布尔巴基学派的影响, 在拓扑学领域取得了举世瞩目的成果, 晚年致力于数学定理机器证明的研究, 并提出了数学机械化纲领. 吴先生开创的定理机器证明 “吴方法” 主要基于多项式代数方法, 依赖符号计算和数值计算, 有完备的算法, 但算法复杂性高, 在变量或参数过多的情况下, 受硬件物理条件限制. 形式化证明主要基于逻辑的推理, 人机交互, 可将人的智慧和机器的智能结合起来. 将计算机证明辅助工具 Coq 和基于符号数值混合计算的多项式代数方法结合, 是非常有吸引力的研究课题. 实现拓扑学的机器证明也是吴文俊院士的一个宿愿.

本书利用交互式定理证明工具 Coq, 实现 Morse-Kelley 公理化集合论形式化系统, 包括对该体系中 8 个公理 (含选择公理) 和 1 个公理图示以及全部 181 条定义或定理的 Coq 描述, 其中构造了序数和基数, 定义了非负整数, 把 Peano 公设当作定理, 可以迅速而自然地给出一个数学基础, 摆脱了明显的悖论. 这是 Morse-Kelley 公理化集合论系统的首次形式化实现. 在 Morse-Kelley 公理化集合论形式化系统下, 作为应用, 给出选择公理与它的几个著名等价命题间等价性的机器证明, 这些命题包括 Tukey 引理、Hausdorff 极大原则、极大原则、Zorn 引理、良序原则及 Zermelo 假定等. 在我们开发的系统中, 全部定理无例外地给出 Coq 的机器证明代码, 所有形式化过程已被 Coq 验证, 并在计算机上运行通过, 体现了基于 Coq 的数学定理机器证明具有可读性和交互性的特点, 其证明过程规范、严谨、可靠. 该系统可方便地应用于拓扑学和代数学理论的形式化构建.

2017 年 5 月 7 日晨, 吴文俊院士因病去逝. 2019 年 5 月 12 日是吴文俊院士 100 周年诞辰的日子. 谨以本书纪念吴文俊院士 100 周年诞辰!

作者水平有限, 书中不当之处在所难免, 挚诚欢迎批评指正, 以期今后改进.

致谢

首先感谢中国科学院成都分院张景中院士、杨路教授和西安交通大学徐宗本院士. 张先生和杨先生自 20 世纪 80 年代即追随吴文俊院士开展数学定理机器证明的研究, 在可读性证明、数值符号混合计算及不等式机器证明等方面做出突出成就. 90 年代, 作者分别在四川大学和北京大学攻读硕士与博士研究生学位期间, 即关注到两位先生在定理机器证明中的相关成果, 并有意识地将数学机械化的思想应用于控制理论的研究, 做出一些成果, 后来又有幸与杨先生合作多年, 受益良多. 经张景中院士和徐宗本院士推荐, 作者 2017 年获吴文俊人工智能自然科学奖. 在此谨向三位先生致以诚挚的谢意.

华东师范大学何积丰院士团队早在 2009 年即在国内引进介绍了证明辅助工具

Coq, 并进行了程序验证方面的深入研究. 作者曾与该团队曾振柄教授 (现在上海大学)、朱惠彪教授、吴敏副教授、杨争峰副教授及赵世忠博士等进行有益的讨论.

作者正在或曾经工作学习的北京邮电大学电子工程学院、华东师范大学计算机与软件学院及伊犁师范大学数学与统计学院的校、院领导, 他们创造了良好的学术氛围, 对本书有极大的支持, 深表谢意.

感谢作者的良师益友北京大学王龙教授, 伊犁师范大学汤建刚教授、梁崇民老师, 西安电子科技大学段振华教授, 以及北京邮电大学忻向军教授、中国矿业大学杨克虎教授、中国科学院自动化研究所关强副研究员, 多年来作者曾与他们一起探讨学术问题, 他们都给予了作者热情的支持和帮助.

本书合作者孙天宇和付尧顺都是在读博士研究生, 在撰写本书的多次深入讨论中, 他们对数学思想的领会, 对科学理性、美感的感悟, 都有了极大的提高; 他们为实现本书中的证明代码付出了艰辛的努力, 对程序完成过程中的酸甜苦辣, 个中滋味, 体会深刻. 相信读者在理解、运行本书定理机器证明的过程中会有切身的认识.

曾在或正在本课题组学习的一批研究生：束润东、郑旭、柴干、傅凡、袁婧、范一凡、郭礼权、刘雅静、席文琦、李昱辰、刘洋、于畅、赵保强、郭达凯 (北京邮电大学)；糟晓燕、刘佳、吕红伟 (伊犁师范大学)；杜慧娟、陈德鹏、倪谢俊 (华东师范大学), 以及远在美国康奈尔大学求学的郁博晨博士等, 他们在提供材料、验证算例及校对文稿等方面给予了帮助, 一并致谢.

特别感谢张景中院士推荐本书给 “数学机械化丛书” 编委会, 感谢丛书编委会主编、中国科学院数学与系统科学研究院高小山研究员针对本书可读性方面提出的有益建议, 并同意将本书收入该丛书系列.

特别感谢责任编辑王丽平、加工编辑李香叶、排版乔秀琴, 在本书立项、排版、校稿的过程中, 她们都花费了很多心血; 感谢科学出版社鄢德平、赵彦超以及商务印书馆乔永, 他们一直关注和支持本书的出版. 他们坚实的专业知识以及一丝不苟的工作作风和实干精神, 令人敬佩!

感谢在撰写本书过程中所有曾给予作者支持和鼓励的同志和单位, 希望本书的出版将无愧于这些支持和帮助.

最后, 感谢国家自然科学基金及华罗庚–吴文俊数学出版基金对本课题的资助.

郁文生
2019 年 3 月

符 号 汇 集

逻辑符号

$\sim$	否定 (非)
$\vee$	析取 (或)
$\wedge$	合取 (与)
$\Longrightarrow$	蕴涵
$\Longleftrightarrow$	等价

量词符号

$\exists$	存在量词
$\forall$	全称量词

基本数学常项符号

$=$	等词
$\in$	属于
$\{\cdots:\cdots\}$	类

集合符号

x 是集 $\Longleftrightarrow (\exists y, x \in y)$

$x \cup y = \{z : z \in x \vee z \in y\}$

$x \cap y = \{z : z \in x \wedge z \in y\}$

$x \notin y \Longleftrightarrow (\sim (x \in y))$

$\neg\, x = \{y : y \notin x)$

$x \sim y = x \cap (\neg\, y)$

$x \neq y \Longleftrightarrow \sim (x = y)$

$0 = \{x : x \neq x\}$

$\mathcal{U} = \{x : x = x\}$

$\bigcup x = \{z : \exists y, z \in y \wedge (y \in x)\}$

$\bigcap x = \{z : \forall y, y \in x \Longrightarrow z \in y\}$

$x \subset y \Longleftrightarrow (\forall z, z \in x \Longrightarrow z \in y)$

$x \subsetneq y \Longleftrightarrow x \subset y \wedge x \neq y$

$2^x = \{y : y \subset x\}$

$\{x\} = \{z : x \in \mathcal{U} \Longrightarrow z = x\}$

$\{xy\} = \{x\} \cup \{y\}$

$(x, y) = \{\{x\}\{xy\}\}$

z 的 1^{st} 坐标 $= \bigcap\bigcap z$

z 的 2^{nd} 坐标 $= (\bigcap\bigcup z) \cup ((\bigcup\bigcup z) \ \sim (\bigcup\bigcap z))$

r 是关系 $\Longleftrightarrow (\forall z \in r, \exists x, \exists y, z = (x, y))$

$r \circ s = \{u : \exists x,\ \exists y,\ \exists z,\ u = (x, z),\ (x, y) \in s \wedge (y, z) \in r\}$

$r^{-1} = \{(x, y) : (y, x) \in r\}$

f 是函数 $\Longleftrightarrow f$ 是关系 $\wedge (\forall x, \forall y, \forall z, ((x, y) \in f \wedge (x, z) \in f) \Longrightarrow y = z)$

f 的定义域 $= \{x : \exists y, (x, y) \in f\}$

f 的值域 $= \{y : \exists x, (x, y) \in f\}$

$f(x) = \bigcap\{y : (x, y) \in f\}$

$x \times y = \{(u, v) : u \in x \wedge v \in y\}$

$y^x = \{f : f$ 是函数 $\wedge (f$ 的定义域 $= x) \wedge (f$ 的值域 $\subset y)\}$

f 在 x 上 $\Longleftrightarrow f$ 是函数 $\wedge\ x = f$ 的定义域

f 到 $y \Longleftrightarrow f$ 是函数 $\wedge\ f$ 的值域 $\subset y$

f 到 y 上 $\Longleftrightarrow f$ 是函数 $\wedge\ f$ 的值域 $= y$

$xry \ \Longleftrightarrow \ (x, y) \in r$

r 连接 $x \Longleftrightarrow \forall u \in x,\ \forall v \in x,\ urv \vee vru \vee u = v$

r 在 x 中是传递的 $\Longleftrightarrow (\forall u \in x,\ \forall v \in x,\ \forall w \in x,\ urv \wedge vrw \Longrightarrow urw)$

r 在 x 中是非对称的 $\Longleftrightarrow (\forall u \in x,\ \forall v \in x,\ urv \Longrightarrow \sim vru)$

z 是 x 的 r- 首元 $\Longleftrightarrow z \in x \wedge (\forall y \in x \Longrightarrow \sim yrz)$

r 良序 $x \Longleftrightarrow r$ 连接 $x \wedge (\forall y \subset x \wedge y \neq 0 \Longrightarrow \exists z,$ z 是 y 的 r- 首元)

y 是 x 的 r-截片 $\Longleftrightarrow y \subset x \wedge r$ 良序 $x \wedge (\forall u \in x,\ \forall v \in y,\ urv \Longrightarrow u \in y)$

f 是 r-s 保序的 $\Longleftrightarrow f$ 是函数, r 良序 f 的定义域, s 良序 f 的值域 $\wedge (\forall u \in f$ 的定义域, $\forall v \in f$ 的定义域, $urv \Longrightarrow f(u)sf(v))$

f 是 1-1 函数 $\Longleftrightarrow f$ 与 f^{-1} 都是函数

f 在 x 和 y 中 r-s 保序 $\Longleftrightarrow r$ 良序 x, s 良序 y, f 是 r-s 保序, f 的定义域是 x 的 r-截片, f 的值域是 y 的 s-截片

$E = \{(x, y) : x \in y\}$

x 是充满的 $\Longleftrightarrow (y \in x \Longrightarrow y \subset x)$

x 是序 $\Longleftrightarrow E$ 连接 $x \wedge x$ 是充满的

$R = \{x : x$ 是序$\}$

x 是序数 $\Longleftrightarrow x \in R$

$x < y \Longleftrightarrow x \in y$

$x \leqslant y \Longleftrightarrow x \in y \vee x = y$

$x + 1 = x \cup \{x\}$

$f|x = f \cap (x \times \mathcal{U})$

x 是整数 $\Longleftrightarrow x$ 是序 $\wedge\ E^{-1}$ 良序 x

x 是 y 的一个 E-末元 $\Longleftrightarrow x$ 是 y 的一个 E^{-1}-首元

$\omega = \{x : x$ 是整数$\}$

c 是选择函数 $\Longleftrightarrow c$ 是函数 $\wedge\ (\forall x \in c$ 的定义域, $c(x) \in x)$

n 是套 $\Longleftrightarrow (x \in n \wedge y \in n \Longrightarrow x \subset y \vee y \subset x)$

$x \approx y \Longleftrightarrow (\exists\ f,\ f$ 是 1-1 函数, f 的定义域 $= x$, f 的值域 $= y)$

x 是基数 $\Longleftrightarrow x$ 是序数 $\wedge\ ((\forall y, y \in R, y < x) \Longrightarrow \sim (x \approx y))$

$C = \{x : x$ 是基数$\}$

$P = \{(x, y) : x \approx y \wedge y \in C\}$

x 是有限的 $\Longleftrightarrow P(x) \in \omega$

$\max[x, y] = x \cup y$

$\ll\ = \{z : \exists (u, v) \in R \times R, \exists (x, y) \in R \times R, z = ((u, v), (x, y)),$
$\quad (\max[u, v] < \max[x, y]) \vee (\max[u, v] = \max[x, y] \wedge u < x)$
$\quad \vee (\max[u, v] = \max[x, y] \wedge u = x \wedge v < y)\}$

目　　录

第 1 章　引　　言

人工智能是当前科技发展的热点和前沿方向, 夯实人工智能基础理论尤为重要, 数学定理机器证明是人工智能基础理论的深刻体现[1,3-5,9,24-29,32,46,52,53,61,63-67,78,80,81,85-88,93].

数学定理的计算机形式化证明, 近年来随着计算机科学的迅猛发展, 特别是证明辅助工具 Coq、Isabelle 及 HOL 等[3,8,17,30,33,35,51,54,78,79] 的出现, 取得了长足的进展[3-5,24-29,32,52,53,61,63-67,78,87,88]. Coq 是一个交互式定理证明与程序开发系统平台, 是一个用于描述定理内容、验证定理证明的计算机工具. 这些定理可能涉及普通数学、证明理论或程序验证等[2,3,8,15,17,35,43,44,54,78] 方面. 在推理和编程方面, Coq 都拥有足够强大的表达能力, 从构造简单的项, 执行简单的证明, 直至建立完整的理论, 学习复杂的算法等, 对学习者的能力有着不同层次的需求[8,17,35,54].

Coq 是一个交互式的编译环境[8,17,35,54], 用户以人机对话的方式一问一答, 用户可以边设计、边修改, 使证明中的错误及时得到补证. Coq 系统的基本原理是归纳构造演算[8,17,35,54], 是一个形式化系统. Coq 支持自动推理程序. Coq 通过命令式程序进行逻辑推导, 可以利用已证命题进行自动推理. Coq 中的归纳类型扩展了传统程序设计语言中有关类型定义的概念, 类似于大多数函数式程序设计语言中的递归类型定义[8,17,35,54]. Coq 有一支强大的全职研发队伍, 支持开源.

目前, Coq 已成为数学定理证明与计算逻辑领域[2,3,15,43,44,54,78] 的一个重要工具. 2005 年, 国际著名计算机专家 Gonthier 和 Werner 成功基于 Coq 给出了著名的 “四色定理” 的计算机证明[24], 进而, Gonthier 又经过六年努力, 于 2012 年完成对 “有限单群分类定理” 的机器验证 (该证明过程约有 4000 个定义和 13000 条定理, 约 150000 行 Coq 代码)[4,25,26], 2015 年, Hales 等又完成了对 “Kepler 猜想” 的机器验证[32], 使得证明辅助工具 Coq 在学术界得到广泛认可. Wiedijk[79] 指出, 全球各相关研究团队已经或计划完成包括 Gödel 第一不完备性定理、Jordan 曲线定理、素数定理以及 Fermat 大定理等在内的 100 个著名数学定理的计算机形式化证明. 这些成果使得证明辅助工具 Coq 在学术界的影响日益增强.

另一方面, 数学定理的形式化证明, 必然涉及国际著名的布尔巴基 (Bourbaki) 学派[10-13]. 布尔巴基是一群以法国人为主的数学家的共同名字, 他们的思想对 20 世纪中叶以来的数学发展具有深刻影响[39,42,75]. 该学派提出数学结构的概念,

并用此概念统一现代数学 [10, 39, 42, 75]. 按照布尔巴基学派的观点, 数学中有三大母结构, 即序结构[13]、代数结构[11] 和拓扑结构[12]. 基于这三大结构相互交融形成了现代数学的主体内容. 利用交互式定理证明工具 Coq, 可以完整构建这三大母结构的机器证明系统, 在此方面国内外都开始进行了一些有益的研究工作[3,5,27-29,61,65-67,78,87,88].

对于数学理论的形式化来说, 公理化集合论的形式化实现尤为重要, 因为它是现代数学的基础 [13, 39, 42, 45, 75]. 19 世纪末与 20 世纪初, 朴素集论[14, 31, 36, 60, 71, 82] 中一些悖论 [14, 34, 36, 39, 42, 75, 77] 的发现, 使集合论之公理化研究成为必要[7,18-20,38,41,50,68,74,92]. 公理化集合论的出发点就是给出一组集合应该满足的公理, 在此基础上研究集合的性质. 集合论里普遍采用的公理体系是 ZFC 系统和 NBG 系统, 前者由 Zermelo-Fraenkel (ZF) 集合论公理加上选择公理 (Axiom of Choice, AC) 构成, 后者以 von Neumann, Bernays 和 Gödel 三位数学家名字的首字母命名, 其中后者提出 "真类" 的概念, 真类不能作为任何集合或类的元素. 详细介绍可参见文献 [7, 18-20, 37, 38, 48, 49, 55-58, 70, 74, 83, 84, 89, 90, 92, 95].

选择公理是集合论里有关映射存在性的一条公理, 最早于 1904 年由 Zermelo 提出, 并用于对良序定理的证明 [89]. 选择公理在现代数学中有很重要的作用, 与许多深刻的数学结论有着十分密切的联系. 没有选择公理, 我们甚至无法知道两个集合能否比较元素的多少、任何一族非空集的积是否非空、线性空间是否一定有一组基、任何一族紧致空间的积是否紧致、环是否一定有极大理想 [55, 84, 90]. 选择公理有几十甚至上百个等价的形式 [37, 55-57, 83].

2015 年以来, 本书作者团队分别在北京邮电大学和伊犁师范大学开始了基于 Coq 的 "公理化集合论"、"近世代数基础" 和 "一般拓扑学" 形式化系统的研发, 对布尔巴基学派强调的现代数学三大母结构形式化系统的机器实现进行了有意义的探索尝试 [61, 63–67, 87, 88].

我们基于 Coq 开发的 "公理化集合论" 形式化系统, 以 Kelley[41] 的 Morse-Kelley 公理化集合论体系为依据. Morse-Kelley (MK) 公理化集合论体系的思想最早由王浩提出 [76], 1955 年 Kelley[41] 正式发表, 此后在 1965 年由 Morse[50] 给出了自己的版本. Kelley[41] 明确指出, 他的 "公理体系是 Skolem 和 Morse 体系的变形, 且更接近于由 Gödel 所系统叙述的 Hilbert-Bernays-von Neumann 体系", "同时构造了序数和基数, 定义了非负整数, 并把 Peano 公设当作定理给予了证明", 并且实数也可以由 "整数类是一个集" 和 "利用归纳法在整数上定义一个函数是可能的" 事实, 以及 Peano 公设和无限性公理来构造 [41, 45]. 该公理化可 "用来迅速而又自然地给出一个数学基础, 其中摆脱了明显的悖论. 由于这个缘故, 有限的公理体系被遗弃, 而把整个理论建筑在 8 个公理和 1 个公理图示之上 (也就是说, 在某种指定的形式下的一切语句都被认作公理)"[41]. 这充分说明了 Morse-Kelley 公理体系

的简洁、严谨和优美, 而且在此基础上, 现代数学的拓扑学和代数学理论可方便、快速地形式化构建, 事实上, Kelley 也正是在其名著《一般拓扑学》[41] 中发表他的公理集合论体系的. 该公理体系相当于承认存在比集合更广的类, 与 ZFC 公理体系和 NBG 公理体系无矛盾 [7,18-20,37,38,48,49,55-58,70,74,83,84,89,90,92,95]. NBG 系统是 ZF 系统的保守扩展. 而 MK 系统是 ZFC 系统的一个真扩展. Mendelson[48], Monk[49] 及 Rubin[58] 等数学家均认为, MK 系统较 ZFC 系统和 NBG 系统应用起来更为便利.

本书利用交互式定理证明工具Coq, 给出 Morse-Kelley 公理化集合论体系 [34,41,50,76] 的形式化实现, 包括对 Kelley[41] 给出该体系中 8 个公理 (含选择公理) 和 1 个公理图示以及全部 181 条定义或定理的 Coq 描述. 这是 Morse-Kelley 集合论的首次形式化实现. 作为应用, 给出选择公理与它的几个著名等价命题间等价性的机器证明, 这些命题包括 Tukey 引理、Hausdorff 极大原则、极大原则、Zorn 引理、良序定理及 Zermelo 假定等. 在我们开发的系统中, 全部定理无例外地给出 Coq 的机器证明代码, 所有形式化过程已被 Coq 验证, 并在计算机上运行通过, 体现了基于 Coq 的数学定理机器证明具有可读性和交互性的特点, 其证明过程规范、严谨、可靠. 顺便指出, ZFC 公理集合论和 NBG 公理集合论也可开发相应的形式化实现系统, 例如, 基于 Coq 的可参见文献 [27-29], 基于 Isabelle 的可参见文献 [52,53].

本书结构安排如下：第 1 章给出背景知识的介绍；第 2 章给出基本 Coq 指令清单和预备知识；第 3 章给出 Morse-Kelley 公理化集合论体系的完整形式化构建, 书中一般在相关定义、公理和定理的人工描述之后, 一并给出其精确的 Coq 描述代码, 而所有定理的证明过程均通过 Coq 代码完成；第 4 章给出选择公理与它的几个著名等价命题间等价性的机器证明；第 5 章总结我们的结论并给出相关注记.

第 2 章　基本 Coq 指令清单和预备知识

在给出 Morse-Kelley 公理化集合论体系的完整形式化构建之前, 本章给出一些证明过程中会用到的基本 Coq 指令及它们的简要用法说明, 这些指令在使用过程中是容易理解的, 各指令的详细功能也可参见 [8,17,35,54]. 指令清单见表 2.1.

表 2.1　书中涉及 Coq 常用指令简表

Coq 指令	用法
intro/intros	引入目标中的条件
unfold	将定义展开
rewrite H	将 H 的右边当作左边代入目标
rewrite <− H	将 H 的左边当作右边代入目标
elim	消去, 对条件进行分解
apply	应用指定条件到证明目标
destruct	拆掉指定条件中的或和与
split	拆分目标中间的与
left	证明或左边的目标
right	证明或右边的目标
generalize	引入假设条件
assert	指定一个新的假设条件并证明
auto	自动重复使用 intros, apply, rewrite 等简单策略

一些初等逻辑的实用知识是必要的, 同时, 我们遵循古典二值逻辑：任何命题只允许“真”或“假”的情形而没有其他情形, 并始终默认“**排中律**”成立. 虽然并不需要熟悉形式逻辑理论, 正如 Kelley[41] 指出,“无论如何, 对数学体系本质的理解 (在技术意义下) 有助于弄清和推进某些研讨”. 但为完整起见, 我们也列出常用的初等逻辑运算符号以及基本的逻辑性质, 这里, 采用了文献 [23] 中的一些表述方式.

用通用的初等逻辑运算符号“$\sim$”表示**否定**, 亦即关系词“**非**”; “$\vee$”表示**逻辑析取**, 亦即关系词“**或**”.

用通用的初等逻辑运算符号“$\wedge$”表示**逻辑合取**, 亦即关系词“**与**”.

符号“$\Longrightarrow$”表示**蕴含**. 符号“$\Longleftrightarrow$”表示“**等价**”.

符号“$\exists$”表示**存在量词**(符号“$\exists!$”表示**存在唯一**). 符号“$\forall$”表示**全称量词**.

在上述记号中,“$\wedge$”和“$\Longrightarrow$”可由“$\vee$”和“$\sim$”给出;“$\Longleftrightarrow$”可由“$\Longrightarrow$”和“$\wedge$”给出;“$\forall$”可由“$\sim$”和“$\exists$”给出 [23]. 事实上

$$(A \wedge B) \quad \text{即} \quad (\sim ((\sim A) \vee (\sim B))),$$
$$(A \Longrightarrow B) \quad \text{即} \quad (B \vee (\sim A)),$$
$$(A \Longleftrightarrow B) \quad \text{即} \quad (A \Longrightarrow B) \wedge (B \Longrightarrow A),$$
$$((\forall x) A) \quad \text{即} \quad (\sim ((\exists x)(\sim A))),$$

这里, A 和 B 是命题, x 是变元. 下面列出常用的基本逻辑性质, 在文献 [23] 中这些逻辑性质是以**逻辑公理** (Logical Axioms)和**逻辑重言式** (Logical Tautologies)的形式给出的, 其中 A, B 和 S 均为命题.

AL1　$(A \vee A) \Longrightarrow A$.

AL2　$A \Longrightarrow (A \vee B)$.

AL3　$(A \vee B) \Longrightarrow (B \vee A)$.

AL4　$(A \Longrightarrow B) \Longrightarrow ((A \vee S) \Longrightarrow (B \vee S))$.

TL1　$((A \Longrightarrow B) \wedge (B \Longrightarrow S)) \Longrightarrow (A \Longrightarrow S)$.

TL2　$A \Longrightarrow A$.

TL3　$A \Longrightarrow (\sim (\sim A))$.

TL4　$(A \Longrightarrow B) \Longleftrightarrow ((\sim B) \Longrightarrow (\sim A))$.

TL5　$(A \wedge B) \Longrightarrow A, (A \wedge B) \Longrightarrow B$.

TL6　$((A \vee B) \wedge (A \Longrightarrow S) \wedge (B \Longrightarrow S)) \Longrightarrow S$.

文献 [23] 中涉及命题中变元代换的基本逻辑性质不再列出, 因为在我们开发的 Morse-Kelley 公理集合论形式化系统中, 命题中变元代换是通过分类公理图式[41] 来实现的, 在布尔巴基的相关著作中是通过定义所谓的 "**Hilbert 操作 (the Hilbert Operation)**" 及增加基本逻辑公理和重言式 [10-13,23] 来实现的.

上述基本逻辑概念及性质都已在 Coq 的标准逻辑库 "Classical" 中实现, 事实上, 这些基本逻辑性质或者本身即 Coq 标准逻辑库 "Classical" 中的引理, 或者可由 "Classical" 中的引理直接验证, 我们可以直接调用. Coq 标准逻辑库 "Classical" 中的逻辑理论已经足够我们的系统使用了.

在 Coq 标准逻辑库 "Classical" 中, "**排中律**" 是以公理形式描述的, 其 Coq 代码如下：

```
Axiom classic : forall P:  Prop, P \/ ~ P.
```

排中律也可以定义的形式给出, 其 Coq 代码如下：

```
Definition excluded_middle := forall P: Prop, P \/ ~ P.
```

另外, 我们在系统中还增加了下面三条简单的逻辑性质, 因为, 它们在一些证明中要反复应用, 作为性质调用要方便得多, 也会使机器证明代码更为简洁, 不妨

称之为 “引理 x”、“引理 y” 和 “引理 z”.

引理 x　A 是命题, 则 $A \Longrightarrow (A \wedge A)$.

引理 y　A 和 B 都是命题, 则 A 和 $B \Longrightarrow (A \wedge B)$.

引理 z　A 和 B 都是命题, 则 $((A \Longleftrightarrow B)$ 和 $(\sim A)) \Longrightarrow (\sim B)$.

上面的三条引理是显然的, 我们也给出了它们证明验证的代码. 本章预备知识的完整 Coq 代码如下:

```
Require Export Classical.

(* A.0 基本逻辑 *)

Module Property.

Proposition Lemma_x : forall A : Prop, A -> A /\ A.
Proof. intros; split; auto. Qed.

Proposition Lemma_y : forall (A B : Prop), A -> B -> A /\ B.
Proof. intros; split; auto. Qed.

Proposition Lemma_z : forall A B, (A <-> B) -> (∼ A) ->(∼ B).
Proof. unfold not; intros; apply H in H1;
apply H0 in H1; auto. Qed.

Ltac double H := apply Lemma_x in H; destruct H.

Ltac add B H:= apply (Lemma_y _ B) in H; auto.

Notation "∀ x .. y , P" := (forall x, .. (forall y, P) ..)
  (at level 200, x binder, y binder, right associativity,
  format "'[ ' ∀ x .. y ']' , P") : type_scope.

Notation "∃ x .. y , P" := (exists x, .. (exists y, P) ..)
  (at level 200, x binder, y binder, right associativity,
  format "'[ ' ∃ x .. y ']' , P") : type_scope.

Notation "'λ' x .. y , t" := (fun x => .. (fun y => t) ..)
  (at level 200, x binder, y binder, right associativity,
  format "'[ ' 'λ' x .. y ']' , t").

Hint Resolve Lemma_x Lemma_y Lemma_z : set.

End Property.

Export Property.
```

本章代码存为 A_0.v 文件, 命名为 “基本逻辑” 性质模块, 其中出现的一些 Coq 术语是标准的 [8,17,35,54], 在以后各章中也会常用到, 这些术语仅从字面上也可理解其基本含义, 简单列表说明如下, 以后不再特别指出, 可查阅文献 [8,17,35,54] (表 2.2).

表 2.2　书中涉及常用 Coq 术语的基本含义

Coq 术语	基本定义
Require Export	读入库中的文件
Module	开启一个新模块
Proposition	命题
Proof/Qed	证明/证毕
Ltac	组合执行若干命令, 形成一个策略
Notation	引入记号
Hint Resolve	将文件加入库中
End	结束本模块
Export	输出该模块文件

另外, 本章代码的第二行 “(* A.0 基本逻辑 *)”, 是为了便于理解, 插入代码中的注释内容. Notation 命令引入的是一些数学符号, 包括任意量词、存在量词以及标准的λ **演算**表示法 [47].

第 3 章　Morse-Kelley 公理化集合论的形式化系统实现

本章内容是本书的主体, 我们将给出基于 Coq 的 Morse-Kelley 公理化集合论体系的完整形式化系统实现, 包括 Kelley [41] 给出该体系中 8 个公理 (含选择公理) 和 1 个公理图示以及全部 181 条定义或定理的 Coq 描述. 全部定理无例外地给出 Coq 的计算机机器证明代码, 所有形式化过程均被 Coq 验证, 并在计算机上运行通过.

书中一般在相关定义、公理和定理的人工描述之后, 一并给出其精确的 Coq 描述代码, 而所有定理的 Coq 证明代码紧随对应定理描述之后. 为方便起见, 所有定义、公理和定理的序号及人工描述与文献 [41] 一致, 定理证明过程中增加的辅助引理将另行编号. 另外, 为使代码简洁流畅, 在 Coq 证明过程中适当地增加了一些注释内容和批命令策略, 这在上下文中是容易理解的.

3.1　分类公理图式

首先, 定义一个在系统中描述元素和集合概念的 "类", 在 Coq 形式化中定义为 "Class", 用类型 "Type" 表示. 另外, 定义几个基本的常项. 第一个是 "∈", 读作 "属于". 因为本系统中不区分元素和集合的类型, 统一用 "Class" 来表示. 通过 "Notation" 在 Coq 中添加相应数学符号, 增强代码可读性, 形式化定义如下：

```
Require Export A_0.

(* A.1 分类公理图式 *)

Module A1.

(* 定义初始 " 类 (Class) " , 元素和集合的类型都是 Class *)

Parameter Class : Type.

(* ∈: 属于 x∈y : In x y *)

Parameter In : Class -> Class -> Prop.
```

```
Notation "x ∈ y" := (In x y) (at level 10).
```

相等恒用于在逻辑上同一事物的两个名字, 除了通常的相等公理外, 还要假定一个代换规则, 特别在一个定理中, 用一个对象代替与他相等的对象, 结果仍是一个定理, 此外有下面的 “外延公理 I”.

外延公理 I (I Axiom of extent) $x = y \Longleftrightarrow (\forall z, z \in x \Longleftrightarrow z \in y)$.

```
Axiom AxiomI : ∀ (x y: Class), x = y <-> (∀ z: Class,
    z ∈ x <-> z ∈ y).

Hint Resolve AxiomI : set.
```

上面代码中对变元 x, y 和 z 类型的声明是可以省略的, 因为在 Coq 环境中, 它们可由前面引入的 $\in$ 的意义自动识别确认.

有人曾尝试用外延公理 I 作为相等的定义, 从而减去一条公理, 省略所有关于相等的逻辑前提, 这是完全可以办到的. 但是, 对于等式, 这时再也没有无限制的代换规则了, 且必须假设一条公理 [41]: 若 $x \in z, y = x$, 则 $y \in z$.

定义 1 x 是集 $\Longleftrightarrow (\exists y, x \in y)$.

```
Definition Ensemble x : Prop := ∃ y, x ∈ y.

Ltac Ens := unfold Ensemble; eauto.

Ltac AssE x := assert (Ensemble x); Ens.

Hint Unfold Ensemble : set.

End A1.

Export A1.
```

3.2 分类公理图式 (续)

引入第二个常项是分类符号 “$\{\cdots:\cdots\}$”, 读作 “{ 所有 $\cdots$ 的类使得 $\cdots$ }.”

```
Require Export A_1.

(* A.2 分类公理图式续 *)

Module A2.

(* {...:...} *)
```

```
Parameter Classifier : ∀ P: Class -> Prop, Class.

Notation "\{ P \}" := (Classifier P) (at level 0).
```

分类公理图式 II (II Classification axiom-scheme) $\beta \in \{\alpha : P(\alpha)\} \Longleftrightarrow (\beta$ 是集 $\wedge\, P(\beta))$, 这里 $P(\cdot)$ 是一个适定的公式.

```
Axiom AxiomII : ∀ (b: Class) (P: Class -> Prop),
  b ∈ \{ P \} <-> Ensemble b /\ (P b).

Hint Resolve AxiomII : set.

End A2.

Export A2.
```

这里需要说明的是, 每次应用分类公理图式 II 时都因公式 P 的不同而具有不同意义, 这实际包含了无限条的公理, 而 "适定的公式" 是指, 该公式适合 [41]:

(a) 对于下面的每一个用变元替换 "α" 和 "β" 所得的结果是一个公式:

$$\alpha = \beta, \quad \alpha \in \beta.$$

(b) 对于下面的每一个用变元替换 "α" 和 "β" 与用公式替换 "A" 和 "B" 所得的结果是一个公式:

$$A \Longrightarrow B; \quad A \Longleftrightarrow B; \quad \sim A;$$

$$A \wedge B; \quad A \vee B;$$

$$\forall \alpha, A; \quad \exists \alpha, A;$$

$$\beta \in \{\alpha : A\}; \quad \{\alpha : A\} \in \beta; \quad \{\alpha : A\} \in \{\beta : B\}.$$

从 (a) 中的原始公式开始, 按 (b) 中所允许的构造, 递归地构造出来的均为公式.

本系统中所有的公式均为适定的.

定义 1 和分类公理图式 II 对于消除朴素集合论中明显的逻辑悖论是关键的.

可以看到, 利用简单通用的数学逻辑符号 "$\vee$" 和 " $\sim$ ", 加上常项符号 "$\in$"、"$=$" 和 "$\{\cdots : \cdots\}$", 通过若干条公理, 即可构建起整个公理化集合论的形式化系统, 从而为现代数学理论构筑了坚实的基础. 正如文献 [23] 开篇指出, "目的是引进集合和函数的概念. 没有这些概念, 我们在数学上什么也不能做. 反之, 使用这些概念, 我们能够做一切".

3.3 类的初等代数

定义 2 $x \cup y = \{z : z \in x \vee z \in y\}$.

```
Require Export A_2.

(* A.3 类的初等代数 *)

Module A3.

Definition Union x y : Class := \{ λ z, z ∈ x \/ z ∈ y \}.

Notation "x ∪ y" := (Union x y) (at level 65, right associativity).

Hint Unfold Union : set.
```

定义 3 $x \cap y = \{z : z \in x \wedge z \in y\}$.

```
Definition Intersection x y : Class := \{ λ z, z ∈ x
   /\ z ∈ y \}.

Notation "x ∩ y" := (Intersection x y) (at level 60, right
  associativity).

Hint Unfold Intersection : set.
```

定义 2 和定义 3 是说, 类 $x \cap y$ 是 x 与 y 的 “**交**”; 类 $x \cup y$ 是 x 与 y 的 “**并**”.

定理 4 $z \in x \cup y \Longleftrightarrow (z \in x \vee z \in y)$; $\quad z \in x \cap y \Longleftrightarrow (z \in x \wedge z \in y)$.

```
Theorem Theorem4 : ∀ x y z, z ∈ x \/ z ∈ y <-> z ∈ (x ∪ y).
Proof.
  intros; split; intros.
  - apply AxiomII; split; try apply H; destruct H; Ens.
  - apply AxiomII in H; apply H.
Qed.

Theorem Theorem4' : ∀ x y z, z ∈ x /\ z ∈ y <-> z ∈ (x ∩ y).
Proof.
  intros; unfold Intersection; split; intros.
  - apply AxiomII; split; Ens; exists y; apply H.
  - apply AxiomII in H; apply H.
Qed.

Hint Resolve Theorem4 Theorem4' : set.
```

定理 5 $x \cup x = x$; $x \cap x = x$.

```
Theorem Theorem5 : ∀ x, x ∪ x = x.
Proof.
  intros; apply AxiomI; split; intros.
  - apply Theorem4 in H; tauto.
  - apply Theorem4; tauto.
Qed.

Theorem Theorem5' : ∀ x, x ∩ x = x.
Proof.
  intros; apply AxiomI; split; intros.
  - apply Theorem4' in H; tauto.
  - apply Theorem4'; tauto.
Qed.

Hint Rewrite Theorem5 Theorem5' : set.
```

定理 6 $x \cup y = y \cup x$; $x \cap y = y \cap x$.

```
Theorem Theorem6 : ∀ x y, x ∪ y = y ∪ x.
Proof.
  intros; apply AxiomI; split; intro.
  - apply Theorem4 in H; apply Theorem4; tauto.
  - apply Theorem4 in H; apply Theorem4; tauto.
Qed.

Theorem Theorem6' : ∀ x y, x ∩ y = y ∩ x.
Proof.
  intros; apply AxiomI; split; intro.
  - apply Theorem4' in H; apply Theorem4'; tauto.
  - apply Theorem4' in H; apply Theorem4'; tauto.
Qed.

Hint Rewrite Theorem6 Theorem6' : set.
```

定理 7 $(x \cup y) \cup z = x \cup (y \cup z)$; $(x \cap y) \cap z = x \cap (y \cap z)$.

```
Theorem Theorem7 : ∀ x y z, (x ∪ y) ∪ z = x ∪ (y ∪ z).
Proof.
  intros; apply AxiomI; split; intro.
  - apply Theorem4 in H; apply Theorem4; destruct H.
    + apply Theorem4 in H; destruct H; try tauto.
      right; apply Theorem4; auto.
    + right; apply Theorem4; auto.
```

```
  - apply Theorem4 in H; apply Theorem4; destruct H.
    + left; apply Theorem4; auto.
    + apply Theorem4 in H; destruct H; try tauto.
      left; apply Theorem4; tauto.
Qed.

Theorem Theorem7' : ∀ x y z, (x ∩ y) ∩ z = x ∩ (y ∩ z).
Proof.
  intros; apply AxiomI; split; intro.
  - repeat (apply Theorem4' in H; destruct H).
    apply Theorem4'; split; auto; apply Theorem4'; auto.
  - apply Theorem4' in H; destruct H; apply Theorem4'.
    apply Theorem4' in H0; destruct H0; split; auto.
    apply Theorem4'; split; auto.
Qed.

Hint Rewrite Theorem7 Theorem7' : set.
```

这些定理说明交与并在通常意义下的交换律和结合律成立, 而下面是验证分配律.

定理 8 $x\cap(y\cup z)=(x\cap y)\cup(x\cup z);\quad x\cup(y\cap z)=(x\cup y)\cap(x\cup z).$

```
Theorem Theorem8 : ∀ x y z, x ∩ (y ∪ z) = (x ∩ y) ∪ (x ∩ z).
Proof.
  intros; apply AxiomI; split; intros.
  - apply Theorem4; apply Theorem4' in H; destruct H.
    apply Theorem4 in H0; destruct H0.
    + left; apply Theorem4'; split; auto.
    + right; apply Theorem4'; split; auto.
  - apply Theorem4 in H; apply Theorem4'; destruct H.
    + apply Theorem4' in H; destruct H; split; auto.
      apply Theorem4; left; auto.
    + apply Theorem4' in H; destruct H; split; auto.
      apply Theorem4; right; auto.
Qed.

Theorem Theorem8' : ∀ x y z, x ∪ (y ∩ z) = (x ∪ y) ∩ (x ∪ z).
Proof.
  intros; apply AxiomI; split; intro.
  - apply Theorem4'; apply Theorem4 in H.
    destruct H; split; try apply Theorem4; auto.
    + apply Theorem4' in H; tauto.
    + apply Theorem4' in H; tauto.
  - apply Theorem4; apply Theorem4' in H; destruct H.
    apply Theorem4 in H; apply Theorem4 in H0.
    destruct H, H0; auto; right; apply Theorem4'; auto.
```

```
Qed.

Hint Rewrite Theorem8 Theorem8' : set.
```

定义 9 $x \notin y \Longleftrightarrow (\sim (x \in y))$.

```
Definition NotIn x y : Prop := ∼ x ∈ y.

Notation "x ∉ y" := (NotIn x y) (at level 10).

Hint Unfold NotIn : set.
```

定义 10 $\neg\, x = \{y : y \notin x\}$.

```
Definition Complement x : Class := \{ λ y, y ∉ x \}.


Notation "¬ x" := (Complement x) (at level 5, right associativity).

Hint Unfold Complement : set.
```

类 $\neg\, x$ 是 x 的 “**余**”.

定理 11 $\neg\,(\neg\, x) = x$.

```
Theorem Theorem11: ∀ x, ¬ (¬ x) = x.
Proof.
  intros; apply AxiomI; split; intro.
  - apply AxiomII in H; unfold NotIn in H; destruct H.
    assert (z ∈ ¬ x <-> Ensemble z /\ z ∉ x ).
    { split; intros.
      - apply AxiomII in H1; auto.
      - apply AxiomII; auto. }
    apply Lemma_z in H1; auto.
    apply not_and_or in H1; destruct H1; tauto.
  - apply AxiomII; split; Ens; unfold NotIn; intro.
    apply AxiomII in H0; destruct H0; contradiction.
Qed.

Hint Rewrite Theorem11 : set.
```

定理 12 (De Morgan 律) $\neg\,(x \cup y) = (\neg\, x) \cap (\neg\, y)$; $\neg\,(x \cap y) = (\neg\, x) \cup (\neg\, y)$.

```
Theorem Theorem12 : ∀ x y, ¬ (x ∪ y) = (¬ x) ∩ (¬ y).
Proof.
  intros; generalize (Theorem4 x y); intros.
```

```
  apply AxiomI; split; intros.
  - apply AxiomII in H0; destruct H0; unfold NotIn in H1.
    apply Lemma_z with (B:= z ∈ x \/ z ∈ y ) in H1.
    + apply not_or_and in H1; apply Theorem4'; split.
      * apply AxiomII; split; auto; unfold NotIn; tauto.
      * apply AxiomII; split; auto; unfold NotIn; tauto.
    + split; apply H.
  - apply Theorem4' in H0; destruct H0.
    apply AxiomII in H0; apply AxiomII in H1.
    apply AxiomII; split; try tauto.
    destruct H0, H1; unfold NotIn in H2,H3; unfold NotIn.
    apply Lemma_z with (A:= z ∈ x \/ z ∈ y ); auto.
    apply and_not_or; split; auto.
Qed.

Theorem Theorem12' : ∀ x y, ¬ (x ∩ y) =(¬ x) ∪ (¬ y).
Proof.
  intros; generalize (Theorem4' x y); intros.
  apply AxiomI; split; intro.
  - apply AxiomII in H0; unfold NotIn in H0; destruct H0.
    apply Lemma_z with (B:= z ∈ x /\ z ∈ y) in H1.
    + apply Theorem4; apply not_and_or in H1; destruct H1.
      * left; apply AxiomII; split; auto.
      * right; apply AxiomII; split; auto.
    + split; apply H.
  - apply AxiomII; split; Ens.
    unfold NotIn; apply Lemma_z with (A:= z ∈ x /\ z ∈ y); auto.
    apply or_not_and; apply Theorem4 in H0; destruct H0.
    + apply AxiomII in H0; unfold NotIn in H0; tauto.
    + apply AxiomII in H0; unfold NotIn in H0; tauto.
Qed.

Hint Rewrite Theorem12 Theorem12' : set.
```

定义 13 $x \sim y = x \cap (\neg\, y)$.

```
Definition Setminus x y : Class := x ∩ (¬ y).

Notation "x ∼ y" := (Setminus x y) (at level 50, left
  associativity).

Hint Unfold Setminus : set.
```

类 $x \sim y$ 是 x 与 y 之 "**差**", 或者 y 相对于 x 的 "**余**".

定理 14 $x \cap (y \sim z) = (x \cap y) \sim z$.

```
Theorem Theorem14 : forall x y z, x ∩ (y ~ z) = (x ∩ y) ~ z.
Proof.
  intros; unfold Setminus; rewrite Theorem7'; auto.
Qed.

Hint Rewrite Theorem14 : set.
```

文献 [41] 中, 下面定义 15 中首次出现了字符 "$\neq$", 但未加说明, 在定义 85 才给出, 定理 35 中也用到了该字符. 为逻辑上的严密性, 在此补充定义及简单性质如下.

定义(不等于) $x \neq y \Longleftrightarrow \sim (x = y)$.

```
Definition Inequality (x y: Class) : Prop := ~ (x = y).

Notation "x ≠ y" := (Inequality x y) (at level 70).

Corollary Property_Ineq : ∀ x y, (x ≠ y) <-> (y ≠ x).
Proof.
  intros; split; intros; intro; apply H; auto.
Qed.

Hint Unfold Inequality: set. Hint Resolve Property_Ineq: set.
```

定义 15 $0 = \{x : x \neq x\}$.

```
Definition ∅ : Class := \{ λ x, x ≠ x \}.

Hint Unfold ∅ : set.
```

类 0 为 "**空类**", 或者 "**零**".

定理 16 $x \notin 0$.

```
Theorem Theorem16 : ∀ x, x ∉ ∅.
Proof.
  intros; unfold NotIn; intro.
  apply AxiomII in H; destruct H; auto.
Qed.

Hint Resolve Theorem16 : set.
```

定理 17 $0 \cup x = x$; $0 \cap x = 0$.

```
Theorem Theorem17 : ∀ x, ∅ ∪ x = x.
Proof.
```

```
  intros; apply AxiomI; split; intro.
  - apply Theorem4 in H; destruct H; try tauto.
    generalize (Theorem16 z); contradiction.
  - apply Theorem4; tauto.
Qed.

Theorem Theorem17' : ∀ x, ∅ ∩ x =∅.
Proof.
  intros; apply AxiomI; split; intro.
  - apply Theorem4' in H; destruct H; auto.
  - generalize (Theorem16 z); contradiction.
Qed.

Hint Rewrite Theorem17 Theorem17' : set.
```

定义 18 $\mathcal{U} = \{x : x = x\}$.

```
Definition 𝓤 : Class := \{ λ x, x = x \}.

Corollary Property_𝓤 : ∀ x, x ∪ (¬ x) = 𝓤.
Proof.
  intros; apply AxiomI; split; intros.
  - apply AxiomII; split; Ens.
  - apply AxiomII in H; destruct H; apply Theorem4.
    generalize (classic (z ∈ x)); intros; destruct H1; try tauto.
    right; apply AxiomII; split; auto.
Qed.

Hint Unfold 𝓤 : set.
Hint Rewrite Property_𝓤 : set.
```

类 $\mathcal{U}$ 是 "**全域**".

定理 19 $x \in \mathcal{U} \Longleftrightarrow x$ 是集.

```
Theorem Theorem19 : ∀ x, x ∈ 𝓤 <-> Ensemble x.
Proof.
  intros; split; intro.
  - apply AxiomII in H; destruct H; tauto.
  - apply AxiomII; split; auto.
Qed.

Hint Resolve Theorem19 : set.
```

定理 20 $x \cup \mathcal{U} = \mathcal{U}$; $\quad x \cap \mathcal{U} = x$.

```
Theorem Theorem20 : ∀ x, x ∪ 𝒰 = 𝒰.
Proof.
  intros; apply AxiomI; split; intro.
  - apply Theorem4 in H; destruct H; try tauto.
    apply Theorem19; Ens.
  - apply Theorem4; tauto.
Qed.

Theorem Theorem20' : ∀ x, x ∩ 𝒰 = x.
Proof.
  intros; apply AxiomI; split; intro.
  - apply Theorem4' in H; tauto.
  - apply Theorem4'; split; auto.
    apply Theorem19; Ens.
Qed.

Hint Rewrite Theorem20 Theorem20' : set.
```

定理 21 $\neg\, 0 = \mathcal{U}$; $\neg\, \mathcal{U} = 0$.

```
Theorem Theorem21 : ¬ ∅ = 𝒰.
Proof.
  intros; apply AxiomI; split; intros.
  - apply Theorem19; Ens.
  - apply Theorem19 in H; apply AxiomII; split; auto.
    apply Theorem16; auto.
Qed.

Theorem Theorem21' : ¬ 𝒰 = ∅.
Proof.
  intros; apply AxiomI; split; intros.
  - apply AxiomII in H; destruct H.
    apply Theorem19 in H; contradiction.
  - intros; apply AxiomII in H; destruct H.
    apply Theorem19 in H;apply AxiomII in H;destruct H;contradiction.
Qed.

Hint Rewrite Theorem21 Theorem21' : set.
```

定义 22 $\bigcap x = \{z : \forall y, y \in x \Longrightarrow z \in y\}$.

```
Definition Element_I x : Class := \{ λ z, ∀ y, y ∈ x -> z ∈ y \}.

Notation "⋂ x" := (Element_I x) (at level 66).
```

```
Hint Unfold Element_I : set.
```

定义 23 $\bigcup x = \{z : \exists y, z \in y \wedge (y \in x)\}$.

```
Definition Element_U x : Class := \{ λ z, ∃ y, z ∈ y /\ y ∈ x \}.

Notation "⋃ x" := (Element_U x) (at level 66).

Hint Unfold Element_U : set.
```

类 $\bigcap x$ 是 x 的 "**元的交**"; 类 $\bigcup x$ 是 x 的 "**元的并**". 这与定义 2 和定义 3 是不同的.

另外, 关于一个族中元之交 (或并) 的约束变项的记号, 在本系统中不需要. 当然可以引进新的记号表示有 "约束变项族" 中元之交 (或并), 但这需要先将 "约束变项族" 的意义解释清楚.

定理 24 $\bigcap 0 = \mathcal{U}$; $\bigcup 0 = 0$.

```
Theorem Theorem24 : ⋂ ∅ = 𝒰.
Proof.
  intros; apply AxiomI; split; intros.
  - apply Theorem19; Ens.
  - apply AxiomII; apply Theorem19 in H; split; auto.
    intros; generalize (Theorem16 y); contradiction.
Qed.

Theorem Theorem24' : ⋃ ∅ = ∅.
Proof.
  intros; apply AxiomI; split; intro.
  - apply AxiomII in H; destruct H, H0, H0.
    generalize (Theorem16 x); contradiction.
  - generalize (Theorem16 z); contradiction.
Qed.

Hint Rewrite Theorem24 Theorem24' : set.
```

定义 25① $x \subset y \Longleftrightarrow (\forall z, z \in x \Longrightarrow z \in y)$.

```
Definition Included x y : Prop := ∀ z, z ∈ x -> z ∈ y.
Notation "x ⊂ y" := (Included x y) (at level 70).

Hint Unfold Included : set.
```

① 有些书中用符号 "$\subseteq$" 表示 "包含于", 而用符号 "$\subset$" 表示 "真包含于", 与这里的符号稍有区别, 本书中用符号 "$\subset$" 表示 "包含于", 用符号 "$\subsetneq$" 表示 "真包含于", 与文献 [41] 一致.

一个类 x 是 y 的一个 "**子类**", 或者说 "x **被包含于** y" 中, 或者说 "y **包含** x" 当且仅当 $x \subset y$. "$\subset$" 和 "$\in$" 不可混淆, 例如, $0 \subset 0$ 但 $0 \in 0$ 不真.

定理 26 $0 \subset x$; $x \subset \mathcal{U}$.

```
Theorem Theorem26 : ∀ x, ∅ ⊂ x.
Proof.
  intros; unfold Included; intros.
  generalize (Theorem16 z); contradiction.
Qed.

Theorem Theorem26' : ∀ x, x ⊂ 𝒰.
Proof.
  intros; unfold Included; intros; apply Theorem19; Ens.
Qed.

Hint Resolve Theorem26 Theorem26' : set.
```

定理 27 $x = y \iff (x \subset y) \wedge (y \subset x)$.

```
Theorem Theorem27 : ∀ x y, (x ⊂ y /\ y ⊂ x) <-> x = y.
Proof.
  intros; split; intros.
  - destruct H; intros; apply AxiomI; split; auto.
  - rewrite <- H; split; unfold Included; auto.
Qed.

Hint Resolve Theorem27 : set.
```

定理 28 $(x \subset y) \wedge (y \subset z) \Longrightarrow x \subset z$.

```
Theorem Theorem28 : ∀ x y z, x ⊂ y /\ y ⊂ z -> x ⊂ z.
Proof.
  intros; destruct H; unfold Included; auto.
Qed.

Hint Resolve Theorem28 : set.
```

定理 29 $x \subset y \iff x \cup y = y$.

```
Theorem Theorem29 : ∀ x y, x ∪ y = y <-> x ⊂ y.
Proof.
  intros; split; intros.
  - unfold Included; intros; apply AxiomI with (z:= z) in H.
    apply H; apply Theorem4; tauto.
  - apply AxiomI; split; intros.
```

```
    + apply Theorem4 in H0; destruct H0; auto.
    + apply Theorem4; tauto.
Qed.

Hint Resolve Theorem29 : set.
```

定理 30 $x \subset y \Longleftrightarrow x \cap y = x.$

```
Theorem Theorem30 : ∀ x y, x ∩ y = x <-> x ⊂ y.
Proof.
  intros; split; intros.
  - unfold Included; intros; apply AxiomI with (z:= z) in H.
    apply H in H0; apply Theorem4' in H0; tauto.
  - apply AxiomI; split; intros.
    + apply Theorem4' in H0; tauto.
    + apply Theorem4'; split; auto.
Qed.

Hint Resolve Theorem30 : set.
```

定理 31 $x \subset y \Longrightarrow (\bigcup x \subset \bigcup y) \wedge (\bigcap y \subset \bigcap x).$

```
Theorem Theorem31 : ∀ x y, x ⊂ y -> (⋃x ⊂ ⋃y) /\ (⋂y ⊂ ⋂x).
Proof.
  intros; split.
  - unfold Included; intros; apply AxiomII in H0; destruct H0.
    apply AxiomII; split; auto; intros; destruct H1.
    exists x0; split; unfold Included in H; destruct H1; auto.
  - unfold Included in H; unfold Included; intros.
    apply AxiomII in H0; destruct H0; apply AxiomII; split; auto.
Qed.

Hint Resolve Theorem31 : set.
```

定理 32 $x \in y \Longrightarrow (x \subset \bigcup y) \wedge (\bigcap y \subset x).$

```
Theorem Theorem32 : ∀ x y, x ∈ y -> (x ⊂ ⋃y) /\ (⋂y ⊂ x).
Proof.
  intros; split.
  - unfold Included; intros; apply AxiomII; split; Ens.
  - unfold Included; intros; apply AxiomII in H0; apply H0; auto.
Qed.

Hint Resolve Theorem32 : set.
```

为方便应用, 补充 “**真包含**” 定义及几个简单的性质如下.

定义(真包含)[①] $x \subsetneq y \Longleftrightarrow x \subset y \wedge x \neq y.$

```
Definition ProperIncluded x y : Prop := x ⊂ y /\ x ≠ y.


Notation "x ⊊ y" := (ProperIncluded x y) (at level 70).

Corollary Property_ProperIncluded : ∀ x y, x ⊂ y ->
 (x ⊊ y) \/ x = y.
Proof.
  intros.
  generalize (classic (x = y)); intros.
  destruct H0; auto.
  left; unfold ProperIncluded; auto.
Qed.

Corollary Property_ProperIncluded' : ∀ x y, x ⊊ y ->
  ∃ z, z ∈ y /\ z ∉ x.
Proof.
  intros.
  unfold ProperIncluded in H; destruct H.
  generalize (Theorem27 x y); intros.
  apply Lemma_z with (B:= (x ⊂ y /\ y ⊂ x)) in H0; try tauto.
  apply not_and_or in H0; destruct H0; try tauto.
  unfold Included in H0; apply not_all_ex_not in H0; destruct H0.
  apply imply_to_and in H0; Ens.
Qed.

Corollary Property_ProperIncluded'' : ∀ x y,
  x ⊂ y \/ y ⊂ x -> ~ (x ⊂ y) -> y ⊊ x.
Proof.
  intros; destruct H.
  - elim H0; auto.
  - unfold ProperIncluded; split; auto.
    intro; rewrite H1 in H.
    pattern x at 2 in H; rewrite <- H1 in H.
    contradiction.
Qed.

Lemma Property_∅ : ∀ x y, y ⊂ x -> x~ y = ∅ <-> x = y.
Proof.
  intros; split; intros.
  - apply Property_ProperIncluded in H; destruct H; auto.
    apply Property_ProperIncluded' in H; destruct H as [z H], H.
```

① 见本节定义 25 的脚注.

```
    assert (z ∈ (x ∼ y)).
    { unfold Setminus; apply Theorem4'; split; auto.
      unfold Complement; apply AxiomII; split; Ens. }
    rewrite H0 in H2; generalize (Theorem16 z); intros; contradiction.
  - rewrite <- H0; apply AxiomI; split; intros.
    + unfold Setminus in H1; apply Theorem4' in H1.
      destruct H1; unfold Complement in H2.
      apply AxiomII in H2; destruct H2; contradiction.
    + generalize (Theorem16 z); intros; contradiction.
Qed.

Hint Unfold ProperIncluded : set.
Hint Resolve Property_ProperIncluded : set.
Hint Resolve Property_ProperIncluded' : set.
Hint Resolve Property_ProperIncluded'' : set.
Hint Resolve Property_∅: set.

End A3.

Export A3.
```

3.4 集的存在性

子集公理 III (III Axiom of subsets) x 是集 $\Longrightarrow (\exists y,\ y$ 是集$,\ \forall z,\ z \subset x \Longrightarrow z \in y)$.

```
Require Export A_3.

(* A.4 集的存在性 *)

Module A4.

Axiom AxiomIII : ∀ x, Ensemble x ->
  ∃ y, Ensemble y /\ (∀ z, z ⊂ x -> z ∈ y).

Hint Resolve AxiomIII : set.
```

定理 33 x 是集 $\wedge\ (z \subset x) \Longrightarrow z$ 是集.

```
Theorem Theorem33 : ∀ x z, Ensemble x ->
  z ⊂ x -> Ensemble z.
Proof.
  intros; apply AxiomIII in H; destruct H; apply H in H0; Ens.
Qed.
```

```
Hint Resolve Theorem33 : set.
```

定理 34 $0 = \bigcap \mathcal{U}$; $\mathcal{U} = \bigcup \mathcal{U}$.

```
Theorem Theorem34 : ∅ = ⋂ 𝒰.
Proof.
  intros; apply AxiomI; split; intros.
  - generalize (Theorem16 z); contradiction.
  - apply AxiomII in H; destruct H; apply H0.
    apply Theorem19; generalize (Theorem26 z); intros.
    apply Theorem33 in H1; auto.
Qed.

Theorem Theorem34' : 𝒰 = ⋃ 𝒰.
Proof.
  apply AxiomI; split; intros.
  - double H; apply AxiomII in H; destruct H; apply AxiomII.
    split; try auto; generalize (AxiomIII z H); intros.
    destruct H2; destruct H2; exists x; split.
    + apply H3; unfold Included; auto.
    + apply Theorem19; auto.
  - apply AxiomII in H; destruct H; apply Theorem19; auto.
Qed.

Hint Rewrite Theorem34 Theorem34' : set.
```

定理 35 $x \neq 0 \Longrightarrow \bigcap x$ 是集.

```
Lemma Property_NotEmpty : ∀ x, x ≠ ∅ <-> ∃ z, z ∈ x.
Proof.
  intros; assert (x = ∅ <-> ∼ (∃ y, y ∈ x)).
  { split; intros.
    - intro; destruct H0; rewrite H in H0.
      apply AxiomII in H0; destruct H0; case H1; auto.
    - apply AxiomI; split; intros.
      + elim H; exists z; auto.
      + generalize (Theorem16 z); contradiction. }
  split; intros.
  - apply Lemma_z with (B:= ∼ (∃ y, y ∈ x)) in H0; auto.
    apply NNPP in H0; destruct H0; exists x0; auto.
  - apply Lemma_z with (A:=(∼ (∃ y, y ∈ x))); auto.
    destruct H; split; auto.
Qed.

Theorem Theorem35 : ∀ x, x ≠ ∅ -> Ensemble (⋂ x).
```

```
Proof.
  intros; apply Property_NotEmpty in H; destruct H; AssE x0.
  generalize (Theorem32 x0 x H); intros.
  destruct H1; apply Theorem33 in H2; auto.
Qed.

Hint Resolve Property_NotEmpty Theorem35 : set.
```

定义 36 $2^x = \{y : y \subset x\}$.

```
Definition PowerSet x : Class := \{ λ y, y ⊂ x \}.

Notation "pow( x )" := (PowerSet x) (at level 0, right
  associativity).

Hint Unfold PowerSet : set.
```

定理 37 $\mathcal{U} = 2^{\mathcal{U}}$.

```
Theorem Theorem37 : 𝒰 = pow( 𝒰 ).
Proof.
  apply AxiomI; split; intros.
  - apply AxiomII; split; Ens; apply Theorem26'.
  - apply AxiomII in H; destruct H; apply Theorem19; auto.
Qed.

Hint Rewrite Theorem37 : set.
```

定理 38 x 是集 $\Longrightarrow 2^x$ 是集 $\wedge\ (\forall y, y \subset x \Longleftrightarrow y \in 2^x)$.

```
Theorem Theorem38 : ∀ x y,
  Ensemble x -> Ensemble pow(x) /\ (y ⊂ x <-> y ∈ pow(x)).
Proof.
  intros; split.
  - apply AxiomIII in H; destruct H, H.
    assert (pow(x) ⊂ x0).
    { unfold Included; intros; apply AxiomII in H1.
      destruct H1; apply H0 in H2; auto. }
    apply Theorem33 in H1; auto.
  - split; intros.
    + apply Theorem33 with (z:=y) in H; auto.
      apply AxiomII; split; auto.
    + apply AxiomII in H0; apply H0.
Qed.

Hint Resolve Theorem38 : set.
```

定理 39　$\mathcal{U}$ 不是集.

```
Lemma Lemma_N : ~ Ensemble \{ λ x, x ∉ x \}.
Proof.
  generalize (classic (\{ λ x, x ∉ x \} ∈ \{ λ x, x ∉ x \})).
  intros; destruct H.
  - double H; apply AxiomII in H; destruct H; contradiction.
  - intro; elim H; apply AxiomII; split; auto.
Qed.

Theorem Theorem39 : ~ Ensemble 𝒰.
Proof.
  unfold not; generalize Lemma_N; intros.
  generalize (Theorem26' \{ λ x, x ∉ x \}); intros.
  apply Theorem33 in H1; auto.
Qed.

Hint Resolve Lemma_N Theorem39 : set.
```

定理 39 表明全域 $\mathcal{U}$ 确是类而不是一个集. 这里, 先构造了一个类 $R=\{x:x\notin x\}$, 然后利用分类公理图示 II 证明类 R 不是一个集, 此即上面为证定理 39 而引进的引理的结论. 可以看到, 若分类公理图示 II 中不包含 "是集" 的限制, 则导致一个明显的矛盾结果: $R\in R\Longleftrightarrow R\notin R$. 这是著名的 Russell 悖论.

至此, 虽然集的存在性在目前已指明的公理基础上尚不能证明, 但却证明了存在不是集的类.

利用后面的正则性公理将容易推出 $R=\mathcal{U}$, 这也提供了全域 $\mathcal{U}$ 不是集的另一种证法.

定义 40　$\{x\}=\{z:x\in\mathcal{U}\Longrightarrow z=x\}$.

```
Definition Singleton x : Class := \{ λ z, x ∈ 𝒰 ->
  z = x \}.

Notation "[ x ]" := (Singleton x) (at level 0, right
  associativity).

Hint Unfold Singleton : set.
```

"**单点**"x 是 $\{x\}$.

定理 41　x 是集 $\Longrightarrow(\forall y, y\in\{x\}\Longleftrightarrow y=x)$.

```
Theorem Theorem41 : ∀ x, Ensemble x -> (∀ y, y∈[x] <-> y=x).
Proof.
  intros; split; intros.
```

```
  - apply AxiomII in H0; destruct H0; apply H1.
    apply Theorem19 in H; auto.
  - apply AxiomII; split; intros; auto.
    rewrite <- H0 in H; auto.
Qed.

Hint Resolve Theorem41 : set.
```

定理 42 x 是集 $\Longrightarrow \{x\}$ 是集.

```
Theorem Theorem42 : ∀ x, Ensemble x -> Ensemble [x].
Proof.
  intros; double H; apply Theorem33 with (x:= pow(x)).
  - apply Theorem38 with (y:=x) in H0; destruct H0; auto.
  - unfold Included; intros.
    apply Theorem38 with (y:=z) in H0; destruct H0.
    apply H2; apply AxiomII in H1; destruct H1.
    apply Theorem19 in H; apply H3 in H.
    rewrite H; unfold Included; auto.
Qed.

Hint Resolve Theorem42 : set.
```

定理 43 $\{x\} = \mathcal{U} \Longleftrightarrow x$ 不是集.

```
Theorem Theorem43 : ∀ x, [x] = 𝒰 <-> ~ Ensemble x.
Proof.
  split; intros.
  - unfold not; intros; apply Theorem42 in H0.
    rewrite H in H0; generalize Theorem39; contradiction.
  - generalize (Theorem19 x); intros.
    apply Lemma_z with (B:= x∈𝒰) in H; try tauto.
    apply AxiomI; split; intros.
    * apply AxiomII in H1; destruct H1; apply Theorem19; auto.
    * apply AxiomII; split; try contradiction.
      apply Theorem19 in H1; auto.
Qed.

Hint Rewrite Theorem43 : set.
```

由定理 43 容易看出, 定理 42 的逆命题也是成立的. 为方便应用, 补充证明代码如下:

```
Theorem Theorem42' : ∀ x, Ensemble [x] -> Ensemble x.
Proof.
```

```
  intros.
  generalize (classic (Ensemble x)); intros.
  destruct H0; auto; generalize (Theorem39); intros.
  apply Theorem43 in H0; auto.
  rewrite H0 in H; contradiction.
Qed.

Hint Resolve Theorem42' : set.
```

定理 44 x 是集 $\Longrightarrow \bigcap\{x\}=x \wedge \bigcup\{x\}=x$; x 不是集 $\Longrightarrow \bigcap\{x\}=0 \wedge \bigcup\{x\}=\mathcal{U}$.

```
Theorem Theorem44 : ∀ x, Ensemble x -> ⋂ [x] = x /\ ⋃ [x] = x.
Proof.
  intros; generalize (Theorem41 x H); intros.
  split; apply AxiomI.
  - split; intros.
    + apply AxiomII in H1; destruct H1; apply H2; apply H0; auto.
    + apply AxiomII; split; Ens; intros.
      apply H0 in H2; rewrite H2; auto.
  - split; intros.
    + apply AxiomII in H1; destruct H1, H2, H2.
      unfold Singleton in H3; apply AxiomII in H3; destruct H3.
      rewrite H4 in H2; auto; apply Theorem19; auto.
    + apply AxiomII; split; Ens; exists x; split; auto.
      unfold Singleton; apply AxiomII; auto.
Qed.

Theorem Theorem44' : ∀ x, ~ Ensemble x ->
  ⋂ [x] = ∅ /\ ⋃ [x] = 𝒰.
Proof.
  intros; apply Theorem43 in H; split; rewrite H.
  - rewrite Theorem34; auto.
  - rewrite <- Theorem34'; auto.
Qed.

Hint Resolve Theorem44 Theorem44' : set.
```

并集公理 IV (IV Axiom of union) x 是集 $\wedge\ y$ 是集 $\Longrightarrow x\cup y$ 是集.

```
Axiom AxiomIV : ∀ x y,
  Ensemble x /\ Ensemble y -> Ensemble (x ∪ y).

Corollary AxiomIV': ∀ x y,
  Ensemble (x ∪ y) -> Ensemble x /\ Ensemble y.
```

```
Proof.
  intros; split.
  - assert (x ⊂ (x ∪ y)).
    { unfold Included; intros; apply Theorem4; tauto. }
    apply Theorem33 in H0; auto.
  - assert (y ⊂ (x ∪ y)).
    { unfold Included; intros; apply Theorem4; tauto. }
    apply Theorem33 in H0; auto.
Qed.

Hint Resolve AxiomIV AxiomIV' : set.
```

定义 45　$\{xy\} = \{x\} \cup \{y\}$.

```
Definition Unordered x y : Class := [x] ∪ [y].

Notation "[ x | y ]" := (Unordered x y) (at level 0).

Hint Unfold Unordered : set.
```

类 $\{xy\}$ 是一个 "**无序偶**".

定理 46　x 是集 $\wedge$ y 是集 $\Longrightarrow$ $\{xy\}$ 是集 $\wedge$ $(z \in \{xy\} \Longleftrightarrow (z = x \vee z = y))$; $\{xy\} = \mathcal{U} \Longleftrightarrow x$ 不是集 $\vee$ y 不是一个集.

```
Theorem Theorem46 : ∀ x y z,
  Ensemble x /\ Ensemble y ->
  Ensemble [x|y] /\ (z ∈ [x|y] <-> (z=x \/ z=y)).
Proof.
  split; intros; destruct H.
  - apply Theorem42 in H; apply Theorem42 in H0; apply AxiomIV; auto.
  - split; intros.
    + apply AxiomII in H1; destruct H1.
      destruct H2; apply AxiomII in H2; destruct H2.
      * left; apply H3; apply Theorem19; auto.
      * right; apply H3; apply Theorem19; auto.
    + apply AxiomII; split.
      * destruct H1; try rewrite <- H1 in H; auto.
        rewrite <- H1 in H0; auto.
      * destruct H1.
        -- left; apply AxiomII; split; rewrite <- H1 in H; auto.
        -- right; apply AxiomII; split; rewrite <- H1 in H0; auto.
Qed.

Theorem Theorem46' : ∀ x y, [x|y] = 𝒰 <->
```

```
  ∼ Ensemble x \/ ∼ Ensemble y.
Proof.
  unfold Unordered; split; intros.
  - generalize (Theorem43 ([x] ∪ [y])); intros.
    destruct H0; rewrite H in H0.
    assert ([𝒰] = 𝒰); try apply Theorem43; try apply Theorem39.
    apply H0 in H2; rewrite <- H in H2.
    assert (Ensemble([x] ∪ [y]) <-> Ensemble [x] /\ Ensemble [y]).
    { split; try apply AxiomIV; try apply AxiomIV'. }
    apply Lemma_z in H3; auto.
    generalize (not_and_or (Ensemble [x]) (Ensemble [y])); intros.
    apply H4 in H3; destruct H3.
    + assert (Ensemble [x] <-> Ensemble x).
      { split; try apply Theorem42'; try apply Theorem42; auto. }
      apply Lemma_z in H5; auto.
    + assert (Ensemble [y] <-> Ensemble y).
      { split; try apply Theorem42'; try apply Theorem42; auto. }
      apply Lemma_z in H5; auto.
  - destruct H; apply Theorem43 in H; rewrite H; try apply Theorem20.
    generalize (Theorem6 𝒰 [y]); intros; rewrite H0; apply Theorem20.
Qed.

Hint Resolve Theorem46 Theorem46' : set.
```

定理 47 x 是集 $\wedge$ y 是集 $\Longrightarrow (\bigcap\{xy\} = x \cap y \wedge \bigcup\{xy\} = x \cup y)$; x 不是集 $\vee$ y 不是集 $\Longrightarrow (\bigcap\{xy\} = 0 \wedge \bigcup\{xy\} = \mathcal{U})$.

```
Theorem Theorem47 : ∀ x y,
  Ensemble x /\ Ensemble y -> (⋂ [x|y] = x ∩ y) /\ (⋃ [x|y] = x ∪ y).
Proof.
  intros; split; apply AxiomI; intros.
  - split; intros.
    + apply Theorem4'.
      split; apply AxiomII in H0;destruct H0; apply H1;apply Theorem4.
      * left; apply AxiomII; split; try apply H; auto.
      * right; apply AxiomII; split; try apply H; auto.
    + apply Theorem4' in H0; destruct H0.
      apply AxiomII; split; intros; try AssE z.
      apply Theorem4 in H2; destruct H2.
      * apply AxiomII in H2; destruct H2; destruct H.
        apply Theorem19 in H; apply H4 in H; rewrite H; auto.
      * apply AxiomII in H2; destruct H2; destruct H.
        apply Theorem19 in H5; apply H4 in H5; rewrite H5; auto.
  - split; intros.
    + apply AxiomII in H0; destruct H0; destruct H1; destruct H1.
```

```
      apply Theorem4 in H2; apply Theorem4.
      destruct H2; apply AxiomII in H2; destruct H2.
      * left; destruct H; apply Theorem19 in H.
        apply H3 in H; rewrite H in H1; auto.
      * right; destruct H; apply Theorem19 in H4.
        apply H3 in H4; rewrite H4 in H1; auto.
    + apply Theorem4 in H0; apply AxiomII.
      split; destruct H0; try AssE z.
      * exists x; split; auto; apply Theorem4; left.
        apply AxiomII; split; try apply H; trivial.
      * exists y; split; auto; apply Theorem4; right.
        apply AxiomII; split; try apply H; trivial.
Qed.

Theorem Theorem47' : ∀ x y,
  ~ Ensemble x \/ ~ Ensemble y -> (⋂ [x|y] = ∅) /\ (⋃ [x|y] = 𝓤).
Proof.
  intros; split; apply Theorem46' in H; rewrite H.
  - rewrite Theorem34; auto.
  - rewrite <- Theorem34'; auto.
Qed.

Hint Resolve Theorem47 Theorem47' : set.


End A4.

Export A4.
```

3.5 序偶: 关系

定义 48 $(x, y) = \{\{x\}\{xy\}\}$.

```
Require Export A_4.

(* A.5 序偶: 关系 *)

Module A5.
Definition Ordered x y : Class := [ [x] | [x|y]].

Notation "[ x , y ]" := (Ordered x y) (at level 0).

Hint Unfold Ordered : set.
```

类 (x, y) 是一 "**序偶**".

定理 49 (x,y) 是集 $\Longleftrightarrow x$ 是集 $\wedge\ y$ 是集； (x,y) 不是集 $\Longrightarrow (x,y)=\mathcal{U}$.

```
Theorem Theorem49 : ∀ x y,
  Ensemble [x,y] <-> Ensemble x /\ Ensemble y.
Proof.
  intros; split; intro.
  - unfold Ordered in H; unfold Unordered in H.
    apply AxiomIV' in H; destruct H.
    apply Theorem42' in H; auto.
    apply Theorem42' in H; auto.
    apply Theorem42' in H0; auto; split; auto.
    unfold Unordered in H0; apply AxiomIV' in H0.
    destruct H0; apply Theorem42' in H1; auto.
  - destruct H; unfold Ordered; unfold Unordered.
    apply AxiomIV; split.
    + apply Theorem42; auto; apply Theorem42; auto.
    + apply Theorem42; auto; apply Theorem46; auto.
Qed.

Theorem Theorem49' : ∀ x y, ~ Ensemble [x,y] -> [x,y] = 𝓤.
Proof.
  intros; generalize (Theorem49 x y); intros.
  apply Lemma_z with (B:= Ensemble x /\ Ensemble y) in H; try tauto.
  apply not_and_or in H; apply Theorem46' in H; auto.
  generalize Theorem39; intros; rewrite <-H in H1.
  unfold Ordered; apply Theorem46'; auto.
Qed.

Hint Resolve Theorem49 Theorem49' : set.
```

定理 50 x 是集 $\wedge\ y$ 是集 $\Longrightarrow (\bigcup(x,y)=\{xy\}) \wedge (\bigcap(x,y)=\{x\}) \wedge (\bigcup\bigcap(x,y)=x) \wedge (\bigcap\bigcap(x,y)=x) \wedge (\bigcup\bigcup(x,y)=x\cup y) \wedge (\bigcap\bigcup(x,y)=x\cap y)$; x 不是集 $\vee\ y$ 不是集 $\Longrightarrow (\bigcup\bigcap(x,y)=0) \wedge (\bigcap\bigcap(x,y)=\mathcal{U}) \wedge (\bigcup\bigcup(x,y)=\mathcal{U}) \wedge (\bigcap\bigcup(x,y)=0)$.

```
Lemma Lemma50 : ∀ x y,
  Ensemble x /\ Ensemble y -> Ensemble [x] /\ Ensemble [x | y].
Proof.
  intros; apply Theorem49 in H; auto.
  unfold Ordered in H; unfold Unordered in H.
  apply AxiomIV' in H; destruct H.
  apply Theorem42' in H; auto.
  apply Theorem42' in H0; auto.
Qed.
```

```
Theorem Theorem50 : ∀ x y,
  Ensemble x /\ Ensemble y -> (⋃[x,y] = [x|y]) /\ (⋂[x,y] = [x]) /\
  (⋃(⋂[x,y]) = x) /\ (⋂(⋂[x,y]) = x) /\ (⋃(⋃[x,y]) = x∪y) /\
  (⋂(⋃[x,y]) = x∩y).
Proof.
  intros; elim H; intros.
  repeat unfold Ordered; apply Lemma50 in H.
  apply Theorem47 in H; auto; elim H; intros; repeat split.
  - rewrite H3; apply AxiomI; split; intros.
    + apply Theorem4 in H4; elim H4; intros; try tauto.
      apply Theorem4; tauto.
    + apply Theorem4; tauto.
  - rewrite H2; apply AxiomI; split; intros.
    + apply Theorem4' in H4; apply H4.
    + apply Theorem4'; split; auto; apply Theorem4; tauto.
  - rewrite H2; apply AxiomI; split; intros.
    + apply AxiomII in H4; destruct H4, H5, H5.
      apply Theorem4' in H6; destruct H6; apply AxiomII in H6.
      destruct H6; rewrite <- H8; auto.
      apply Theorem19; auto.
    + apply AxiomII; split; Ens; exists x.
      split; auto; apply Theorem4'; split.
      * apply AxiomII; split; auto.
      * apply Theorem4; left; apply AxiomII.
        split; try apply H0; trivial.
  - rewrite H2; apply AxiomI; split; intros.
    + apply AxiomII in H4; destruct H4.
      apply H5; apply Theorem4'; split.
      * apply AxiomII; split; auto.
      * apply Theorem4; left; apply AxiomII; split; auto.
    + apply AxiomII; split; Ens.
      intros; apply Theorem4' in H5. destruct H5.
      apply AxiomII in H5. destruct H5. rewrite H7; auto.
      apply Theorem19; auto.
  - rewrite H3; apply AxiomI; split; intros.
    + apply Theorem4; apply AxiomII in H4; destruct H4, H5, H5.
      apply Theorem4 in H6; destruct H6.
      * apply AxiomII in H6; destruct H6; left; rewrite <- H7; auto.
        apply Theorem19; auto.
      * apply Theorem4 in H6; destruct H6.
        -- apply AxiomII in H6; destruct H6.
           left; rewrite <- H7; auto; apply Theorem19; auto.
        -- apply AxiomII in H6; destruct H6.
           right; rewrite <- H7; auto; apply Theorem19; auto.
```

```
    + apply AxiomII; apply Theorem4 in H4; split.
      * unfold Ensemble; destruct H4.
        -- exists x; auto.
        -- exists y; auto.
      * destruct H4.
        -- exists x; split; auto; apply Theorem4; left.
           apply AxiomII; split; auto.
        -- exists y; split; auto; apply Theorem4; right.
           apply Theorem4; right; apply AxiomII; split; auto.
  - rewrite H3; apply AxiomI; split; intros.
    + apply Lemma_x in H4; elim H4; intros.
      apply AxiomII in H5; apply AxiomII in H6.
      destruct H4; apply Theorem4'; split; auto.
      * apply H5; apply Theorem4; left.
        apply AxiomII; split; auto.
      * apply H6; apply Theorem4; right.
        apply Theorem4; right.
        apply AxiomII; split; auto.
    + apply Theorem4' in H4; destruct H4.
      apply AxiomII; split; Ens.
      intros; apply Theorem4 in H6; destruct H6.
      * apply AxiomII in H6; destruct H6; rewrite H7; auto.
        apply Theorem19; auto.
      * apply AxiomII in H6; destruct H6, H7.
        -- apply AxiomII in H7; destruct H7.
           rewrite H8; auto; apply Theorem19; auto.
        -- apply AxiomII in H7; destruct H7.
           rewrite H8; auto; apply Theorem19; auto.
Qed.

Lemma Lemma50' : ∀ x y,
  ∼ Ensemble x \/ ∼ Ensemble y -> ∼ Ensemble [x] \/ ∼ Ensemble [x|y].
Proof.
  intros; elim H; intros.
  - left; apply Theorem43 in H0; auto.
    rewrite H0; apply Theorem39; auto.
  - right; apply Theorem46' in H; auto.
    rewrite H; apply Theorem39; auto.
Qed.

Theorem Theorem50' : ∀ x y,
  ∼ Ensemble x \/ ∼ Ensemble y -> (⋃(⋂[x,y]) = ∅) /\ (⋂(⋂[x,y])=𝒰)
  /\ (⋃(⋃[x,y]) = 𝒰) /\ (⋂(⋃[x,y]) = ∅).
Proof.
  intros; apply Lemma50' in H; auto.
```

```
  apply Theorem47' in H; destruct H.
  repeat unfold Ordered; repeat split.
  - rewrite H; apply Theorem24'; auto.
  - rewrite H; apply Theorem24; auto.
  - rewrite H0; rewrite <- Theorem34'; auto.
  - rewrite H0; rewrite <- Theorem34; auto.
Qed.

Hint Resolve Theorem50 Theorem50' : set.
```

定义 51 z 的 1^{st} 坐标 $=\bigcap\bigcap z$.

```
Definition First z := ⋂ ⋂ z.

Hint Unfold First : set.
```

定义 52 z 的 2^{nd} 坐标 $=(\bigcap\bigcup z)\cup((\bigcup\bigcup z)\ \sim(\bigcup\bigcap z))$.

```
Definition Second z := (⋂ ⋃ z) ∪ (⋃ ⋃ z) ~ (⋃ ⋂ z).

Hint Unfold Second : set.
```

定理 53 $\mathcal{U}$ 的 2^{nd} 坐标 $=\mathcal{U}$.

```
Lemma Lemma53 : 𝒰 ~ ∅ = 𝒰.
Proof.
  intros; apply AxiomI; split; intros.
  - apply Theorem4' in H; destruct H; auto.
  - apply Theorem4'; split; auto.
    apply AxiomII; split.
    * apply Theorem19 in H; auto.
    * apply Theorem16; auto.
Qed.

Theorem Theorem53 : Second 𝒰 = 𝒰.
Proof.
  intros; unfold Second.
  repeat rewrite <-Theorem34'; auto.
  repeat rewrite <-Theorem34 ; auto.
  rewrite Theorem24'; auto.
  rewrite Lemma53; auto.
  apply Theorem20; auto.
Qed.

Hint Rewrite Theorem53 : set.
```

定理 54 x 是集 $\wedge$ y 是集 $\Longrightarrow ((x,y)$ 的 1^{st} 坐标 $=x) \wedge ((x,y)$ 的 2^{nd} 坐标 $=y)$; x 不是集 $\vee$ y 不是集 $\Longrightarrow ((x,y)$ 的 1^{st} 坐标 $=\mathcal{U}) \wedge ((x,y)$ 的 2^{nd} 坐标 $=\mathcal{U})$.

```
Lemma Lemma54 : ∀ x y,
  (x ∪ y) ~ x = y ~ x.
Proof.
  intros.
  apply AxiomI; split; intros.
  - apply Theorem4' in H; apply Theorem4'.
    destruct H; apply Theorem4 in H; split; auto.
    destruct H; auto; apply AxiomII in H0.
    destruct H0; elim H1; auto.
  - apply Theorem4' in H; apply Theorem4'.
    destruct H; split; auto.
    apply Theorem4; tauto.
Qed.

Theorem Theorem54 : ∀ x y,
  Ensemble x /\ Ensemble y -> First [x,y] = x /\ Second [x,y] = y.
Proof.
  intros; apply Theorem50 in H; auto; split.
  - unfold First; apply H.
  - destruct H, H0, H1, H2, H3; unfold Second.
    rewrite H4; rewrite H3; rewrite H1.
    rewrite Lemma54; auto; unfold Setminus.
    rewrite Theorem6'; auto; rewrite <- Theorem8; auto.
    rewrite Property_𝒰; auto; rewrite Theorem20'; auto.
Qed.


Theorem Theorem54' : ∀ x y,
  ~ Ensemble x \/ ~ Ensemble y -> First [x,y] = 𝒰 /\ Second [x,y]=𝒰.
Proof.
  intros; apply Theorem50' in H; auto; split.
  - unfold First; apply H.
  - destruct H, H0, H1; unfold Second.
    rewrite H2; rewrite H1; rewrite H.
    rewrite Lemma53; auto.
    apply Theorem20; auto.
Qed.

Hint Resolve Theorem54 Theorem54' : set.
```

定理 55 x 是集 $\wedge$ y 是集 $\Longrightarrow ((x,y)=(u,v) \Longleftrightarrow (x=u)\wedge(y=v))$.

```
Theorem Theorem55 : ∀ x y u v,
  Ensemble x /\ Ensemble y -> ([x,y] = [u,v] <-> x = u /\ y = v).
Proof.
  intros; double H.
  apply Theorem49 in H. apply Theorem54 in H0.
  destruct H0; split; intros.
  - rewrite H2 in H; apply Theorem49 in H.
    apply Theorem54 in H; destruct H; split.
    + rewrite <- H2 in H; rewrite <- H0; rewrite H; auto.
    + rewrite <- H2 in H3; rewrite H1 in H3; apply H3.
  - destruct H2; rewrite H2; rewrite H3; trivial.
Qed.

Hint Resolve Theorem55 : set.
```

定理 55 是关于序偶的重要性质. 这个定理的内容相比文献 [41] 的对应描述要更广泛些.

定义 56 r 是关系 $\Longleftrightarrow (\forall z \in r, \exists x, \exists y, z = (x, y))$.

```
Definition Relation r : Prop :=
  ∀ z, z ∈ r -> ∃ x y, z = [x,y].

Hint Unfold Relation: set.
```

一个 "**关系**"是一个类, 它的元为序偶.

定义 57 $r \circ s = \{u : \exists x,\ \exists y,\ \exists z,\ u = (x, z),\ (x, y) \in s \wedge (y, z) \in r\}$.

```
(* { (x,y) : ... } *)

Parameter Classifier_P : (Class -> Class -> Prop) -> Class.

Notation "\{\ P \}\" := (Classifier_P P) (at level 0).

Axiom AxiomII_P : ∀ (a b: Class) (P: Class -> Class -> Prop),
  [a,b] ∈ \{\ P \}\ <-> Ensemble [a,b] /\ (P a b).

Axiom Property_P : ∀ (z : Class) (P: Class -> Class -> Prop),
  z ∈ \{\ P \}\ -> (∃ a b, z = [a,b]) /\ z ∈ \{\ P \}\.

Ltac PP H a b := apply Property_P in H; destruct H as [[a [b H]]];
  rewrite H in *.

Hint Resolve AxiomII_P Property_P: set.

Definition Composition r s : Class :=
```

```
  \{\ λ x z, ∃ y, [x,y] ∈ s /\ [y,z] ∈ r \}\.

Notation "r ∘ s" := (Composition r s) (at level 50, no
  associativity).

Definition Composition' r s : Class :=
   \{ λ u, ∃ x y z, u = [x,z] /\ [x,y] ∈ s /\ [y,z] ∈ r \}.

Hint Unfold Composition Composition' : set.
Hint Resolve AxiomII_P: set.
```

类 $r \circ s$ 是 r 与 s"**合成**".

这里, 为了避免过多的记号, 引进了符号 "$\{(x,y):\cdots\}$": $\{(x,y):\cdots\}=\{u:\exists x,\ \exists z,\ u=(x,z)\wedge\cdots\}$, 于是 $r\circ s=\{(x,z):\exists y,\ (x,y)\in s\wedge(y,z)\in r\}$. 特别注意, 这里只是为了记号方便, 仅对类中的元是有序偶时, 才可应用上述性质和策略.

定理 58 $(r\circ s)\circ t=r\circ(s\circ t)$.

```
Theorem Theorem58 : ∀ r s t,
  (r ∘ s) ∘ t = r ∘ (s ∘ t).
Proof.
  intros; apply AxiomI; split; intros.
  - PP H a b. apply AxiomII_P in H0; destruct H0, H1 as [y H1], H1.
    apply AxiomII_P in H2; destruct H2, H3, H3; apply AxiomII_P; split;
    auto.
    exists x; split; try tauto; apply AxiomII_P; split; Ens.
    AssE [a,y];AssE [y,x];apply Theorem49 in H5;apply Theorem49 in H6.
    destruct H5, H6; apply Theorem49; auto.
  - PP H a b; apply AxiomII_P in H0; destruct H0, H1 as [y H1], H1.
    apply AxiomII_P in H1; destruct H1, H3, H3; apply AxiomII_P; split;
    auto.
    exists x; split; auto; apply AxiomII_P; split; Ens.
    AssE [a,x];AssE [y,b];apply Theorem49 in H5;apply Theorem49 in H6.
    destruct H5, H6; apply Theorem49; Ens.
Qed.

Hint Rewrite Theorem58 : set.
```

定理 59 $r\circ(s\cup t)=(r\circ s)\cup(r\circ t);\quad r\circ(s\cap t)\subset(r\circ s)\cap(r\circ t)$.

```
Theorem Theorem59 : ∀ r s t,
  Relation r /\ Relation s -> r ∘ (s ∪ t) = (r ∘ s) ∪ (r ∘ t) /\
  r ∘ (s ∩ t) ⊂ (r ∘ s) ∪ (r ∘ t).
Proof.
  intros; split.
```

```
  - apply AxiomI; split; intros.
    + PP H0 a b; apply AxiomII_P in H1; destruct H1.
      apply Theorem4.
      destruct H2 as [y H2]; destruct H2.
      apply Theorem4 in H2; destruct H2.
      * left; apply AxiomII_P; split; auto.
        exists y; split; auto.
      * right; apply AxiomII_P; split; auto.
        exists y; split; auto.
    + apply Theorem4 in H0; destruct H0; PP H0 a b; apply AxiomII_P.
      * apply AxiomII_P in H1; destruct H1.
        destruct H2 as [y H2]; destruct H2; split; auto.
        exists y; split; auto; apply Theorem4; try tauto.
      * apply AxiomII_P in H1; destruct H1.
        destruct H2 as [y H2]; destruct H2; split; auto.
        exists y; split; auto; apply Theorem4; try tauto.
  - unfold Included; intros; PP H0 a b.
    apply AxiomII_P in H1; destruct H1.
    destruct H2 as [y H2]; destruct H2.
    apply Theorem4' in H2; apply Theorem4'; split.
    + apply AxiomII_P; split; auto.
      exists y; split; try apply H2; auto.
    + apply AxiomII_P; split; auto.
      exists y; split; try apply H2; auto.
Qed.

Hint Resolve Theorem59 : set.
```

定义 60 $r^{-1} = \{(x, y) : (y, x) \in r\}$.

```
Definition Inverse r : Class := \{\ λ x y, [y,x] ∈ r \}\.

Notation "r⁻¹" := (Inverse r)(at level 5).

Hint Unfold Inverse : set.
```

如果 r 是一个关系, r^{-1} 叫做关于 r 之 “**逆关系**”.

定理 61 r 是关系, $(r^{-1})^{-1} = r$.

```
Lemma Lemma61 : ∀ x y, Ensemble [x,y] <-> Ensemble [y,x].
Proof.
  intros; split; intros.
  - apply Theorem49 in H; auto.
    destruct H; apply Theorem49; auto.
  - apply Theorem49 in H; auto.
```

```
    destruct H; apply Theorem49; auto.
Qed.

Theorem Theorem61 : ∀ r, Relation r -> (r⁻¹)⁻¹ = r.
Proof.
  intros; apply AxiomI; split; intros.
  - PP H0 a b; apply AxiomII_P in H1; destruct H1.
    apply AxiomII_P in H2; apply H2.
  - unfold Relation in H; double H0; apply H in H1.
    destruct H1 as [a [b H1]]; rewrite H1 in *; clear H1.
    apply AxiomII_P; split; Ens; apply AxiomII_P; split; auto.
    apply Lemma61; auto; Ens.
Qed.

Hint Rewrite Theorem61 : set.
```

注意, 定理 61 必须添加 “r 是关系” 这一条件, 除非已默认或约定满足该条件. 若 r 不是关系, 容易举出反例, 结论显然不成立, 例如, 可取 $r=\{(b,a),c\}$, 此时 $r^{-1}=\{(a,b)\}$, 而 $(r^{-1})^{-1}=\{(b,a)\}\neq r$.

定理 62 $(r\circ s)^{-1}=s^{-1}\circ r^{-1}$.

```
Theorem Theorem62 : ∀ r s, (r ∘ s)⁻¹ = (s⁻¹) ∘ (r⁻¹).
Proof.
  intros; apply AxiomI; split; intros.
  - PP H a b; apply AxiomII_P in H0; destruct H0 as [H0 H1].
    apply AxiomII_P; split; auto.
    apply AxiomII_P in H1; destruct H1, H2, H2.
    exists x; split.
    + apply AxiomII_P; split; auto.
      apply Lemma61; Ens; exists r; auto.
    + apply AxiomII_P; split; auto.
      apply Lemma61; Ens.
  - PP H a b; apply AxiomII_P in H0; destruct H0, H1, H1.
    apply AxiomII_P; split; auto.
    apply AxiomII_P in H1; apply AxiomII_P in H2.
    apply AxiomII_P; split.
    + apply Lemma61; auto.
    + exists x; split; try apply H0; try apply H2.
      destruct H1; auto.
Qed.

Hint Rewrite Theorem62 : set.

End A5.
```

```
Export A5.
```

3.6 函　数

定义 63　f 是函数 $\Longleftrightarrow f$ 是关系 $\wedge (\forall x, \forall y, \forall z, ((x,y) \in f \wedge (x,z) \in f) \Longrightarrow y = z)$.

```
Require Export A_5.

(* A.6 函数 *)

Module A6.

Definition Function f : Prop :=
  Relation f /\ (∀ x y z, [x,y] ∈ f /\ [x,z] ∈ f -> y = z).

Hint Unfold Function : set.
```

“**函数**”, 也称为 “**映射**”, 是一个满足单值性的关系.

定理 64　f 是函数 $\wedge\, g$ 是函数 $\Longrightarrow f \circ g$ 是函数.

```
Theorem Theorem64 : ∀ f g,
  Function f /\ Function g -> Function (f ∘ g).
Proof.
  intros; destruct H.
  unfold Function; split; intros.
  - unfold Relation; intros; PP H1 a b; eauto.
  - destruct H1; apply AxiomII_P in H1; apply AxiomII_P in H2.
    destruct H1, H2, H3, H4, H3, H4.
    unfold Function in H, H0; destruct H; destruct H0.
    assert (x0=x1). { apply H8 with x; split; auto. }
    rewrite H9 in H5; apply H7 with x1; split; auto.
Qed.

Hint Resolve Theorem64 : set.
```

定义 65　f 的定义域 $= \{x : \exists y, (x,y) \in f\}$.

```
Definition Domain f : Class := \{ λ x, ∃ y, [x,y] ∈ f \}.

Notation "dom( f )" := (Domain f)(at level 5).

Corollary Property_dom : ∀ x y f,
```

```
  [x,y] ∈ f -> x ∈ dom( f ).
Proof.
  intros; unfold Domain; apply AxiomII; split; eauto.
  AssE [x,y]; apply Theorem49 in H0; apply H0.
Qed.

Hint Unfold Domain : set.
```

定义 66 f 的值域 $=\{y:\exists x,(x,y)\in f\}$.

```
Definition Range f : Class := \{ λ y, ∃ x, [x,y] ∈ f \}.

Notation "ran( f )" := (Range f)(at level 5).

Corollary Property_ran : ∀ x y f,
  [x,y] ∈ f -> y ∈ ran( f ).
Proof.
  intros; apply AxiomII.
  split; eauto; AssE [x,y].
  apply Theorem49 in H0; apply H0.
Qed.

Hint Unfold Range : set.
```

定理 67 $\mathcal{U}$ 的定义域 $=\mathcal{U}$; $\mathcal{U}$ 的值域 $=\mathcal{U}$.

```
Theorem Theorem67 : dom( 𝒰 ) = 𝒰 /\ ran(𝒰 ) = 𝒰.
Proof.
  intros; split; apply AxiomI; split; intros.
  - AssE z; apply Theorem19; auto.
  - apply Theorem19 in H.
    unfold Domain; apply AxiomII; split; auto.
    exists z; apply Theorem19.
    apply Theorem49; split; auto.
  - AssE z; apply Theorem19; auto.
  - apply Theorem19 in H.
    unfold Range; apply AxiomII; split; auto.
    exists z; apply Theorem19.
    apply Theorem49; split; auto.
Qed.

Hint Rewrite Theorem67 : set.
```

定义 68 $f(x)=\bigcap\{y:(x,y)\in f\}$.

```
Definition Value f x : Class := ⋂ \{ λ y, [x,y] ∈ f \}.

Notation "f [ x ]" := (Value f x)(at level 5).

Hint Unfold Value : set.
```

```
(* 值的性质一 *)
```

```
Corollary Property_Value : forall f x,
  Function f -> x ∈ dom( f ) -> [x,f[x]] ∈ f.
Proof.
  intros; unfold Function in H;destruct H as [_ H].
  apply AxiomII in H0; destruct H0, H1.
  assert (x0=f[x]).
  - apply AxiomI; split; intros.
    + apply AxiomII; split; intros; try Ens.
       apply AxiomII in H3; destruct H3.
       assert (x0=y). { apply H with x; split; auto. }
       rewrite <- H5; auto.
    + apply AxiomII in H2; destruct H2 as [_ H2].
       apply H2; apply AxiomII; split; auto.
       AssE [x, x0]; apply Theorem49 in H3; apply H3.
  - rewrite <- H2; auto.
Qed.
```

如果 z 属于 f 的每个元之第二个坐标, 而 f 的第一个坐标是 x, 则 $z \in f(x)$.

类 $f(x)$ 是 f 在 x 处的 “值”, 或者在 f 的映射下 x 的 “象”. 应该注意, 如果 x 是 f 的定义域的一个子集, $f(x)$ 并不等于 $\{y : \exists z, z \in x \wedge y = f(z)\}$.

特别地, 当 f 是一个函数时, 根据单点的定义及定理 44, 容易看出, $f(x)$ 与通常意义下的函数值是一致的.

定理 69 $x \notin f$ 的定义域 $\Longrightarrow f(x) = \mathcal{U}$; $x \in f$ 的定义域 $\Longrightarrow f(x) \in \mathcal{U}$.

```
Theorem Theorem69 : ∀ x f,
  ( x ∉ (dom( f )) -> f[x] = 𝒰 ) /\ ( x ∈ dom( f ) -> (f[x]) ∈ 𝒰 ).
Proof.
  intros; split; intros.
  - assert (\{ λ y, [x,y] ∈ f \} = ∅).
    { apply AxiomI; split; intros.
       apply AxiomII in H0; destruct H0.
       apply Property_dom in H1; contradiction.
       generalize (Theorem16 z); intro; contradiction. }
    unfold Value; rewrite H0; apply Theorem24.
  - assert (\{ λ y, [x,y] ∈ f \} <> ∅).
    { intro.
       apply AxiomII in H; destruct H, H1.
```

```
      generalize (AxiomI \{ λ y : Class,[x, y] ∈ f \} ∅); intro;
      destruct H2.
      apply H2 with x0 in H0; destruct H0.
      assert (x0 ∈ ∅).
      { apply H0; apply AxiomII; split; auto.
        AssE [x, x0]; apply Theorem49 in H5; tauto. }
      eapply Theorem16; eauto. }
    apply Theorem35 in H0; apply Theorem19; auto.
Qed.

Hint Resolve Theorem69 : set.
```

上面的定理并不要求 f 是一个函数.

定理 70 f 是函数 $\Longrightarrow f=\{(x,y):y=f(x)\}$.

```
(* 值的性质二 *)

Corollary Property_Value' : ∀ f x,
  Function f -> f[x] ∈ ran(f) -> [x,f[x]] ∈ f.
Proof.
  intros; apply Property_Value; auto.
  apply AxiomII in H0; destruct H0, H1.
  generalize (classic (x ∈ dom( f))); intros.
  destruct H2; auto; apply Theorem69 in H2; auto.
  rewrite H2 in H0; generalize (Theorem39); intro; contradiction.
Qed.

Theorem Theorem70 : ∀ f,
  Function f -> f = \{\ λ x y, y = f[x] \}\.
Proof.
  intros; apply AxiomI; split; intros.
  - double H0; unfold Function, Relation in H; destruct H.
    apply H in H1; destruct H1 as [a [b H1]]; rewrite H1 in*; clear H1.
    apply AxiomII_P; split; try Ens; apply AxiomI; split; intros.
    + apply AxiomII; split; intros; try Ens.
      apply AxiomII in H3; destruct H3.
      apply Lemma_y with (B:=[a, y] ∈ f) in H0; auto.
      apply H2 in H0; rewrite <- H0; auto.
    + unfold Value, Element_I in H1; apply AxiomII in H1; destruct H1.
      apply H3; apply AxiomII; split; auto; AssE [a,b].
      apply Theorem49 in H4; try apply H4.
  - PP H0 a b; apply AxiomII_P in H1; destruct H1.
    generalize (classic (a ∈ dom( f ))); intros; destruct H3.
    + apply Property_Value in H3; auto; rewrite H2; auto.
    + apply Theorem69 in H3; auto.
      rewrite H3 in H2; rewrite H2 in H1.
```

```
      apply Theorem49 in H1; destruct H1 as [_ H1].
      generalize Theorem39; intro; contradiction.
Qed.

Hint Resolve Theorem70 : set.
```

定理 71　f 是函数 $\wedge$ g 是函数 $\Longrightarrow (f=g \Longleftrightarrow (\forall x, f(x)=g(x)))$.

```
Theorem Theorem71 : ∀ f g,
  Function f /\ Function g -> (f = g <-> ∀ x, f[x] = g[x]).
Proof.
  intros; split; intros; try rewrite H0; trivial.
  destruct H; intros; apply (Theorem70 f) in H;
  apply (Theorem70 g) in H1.
  rewrite H; rewrite H1; apply AxiomI; split; intros.
  - PP H2 a b; apply AxiomII_P in H3; apply AxiomII_P.
    destruct H3; split; auto; rewrite <- H0; auto.
  - PP H2 a b; apply AxiomII_P in H3; apply AxiomII_P.
    destruct H3; split; auto; rewrite -> H0; auto.
Qed.

Hint Resolve Theorem71 : set.
```

如果 $f(x)$ 被定义为以 x 为第一坐标的 f 之元的第二个坐标的并, 定理 71 不真. 因为这时如果 $y \in \mathcal{U}$ 且 $y \notin f$ 的定义域, 则 $f(y)=0$. 而且, 如果 $g=f\cup\{(y,0)\}$, 则对于每个 $x, g(x)=f(x)$, 但是 f 不等于 g [41].

代换公理 V (V Axiom of substitution)　f 是函数 $\wedge$ f 的定义域是集 $\Longrightarrow$ f 的值域是集.

```
Axiom AxiomV : ∀ f,
  Function f -> Ensemble dom(f) -> Ensemble ran(f).

Hint Resolve AxiomV : set.
```

合并公理 VI (VI Axiom of amalgamation)　x 是集 $\Longrightarrow \bigcup x$ 是集.

```
Axiom AxiomVI : ∀ x, Ensemble x -> Ensemble (⋃ x).

Hint Resolve AxiomVI : set.
```

上面的两个公理 ① 进一步描述了所有集的类.

① 这两个公理也可以用一个公理来代替：如果 f 是一个函数同时 f 的定义域是一个集, 则 $\bigcup(f$ 的值域) 是一个集 (施用前面已使用过的约束变元记号, 这便很自然地叙述成：如果 d 是一个集, 同时, 对于中的 $a, x(a)$ 是一个集, 则 $\bigcup\{x(a): a\in d\}$ 是一个集). 要想以此得到公理 V 和公理 VI, 可以进行如下：关于公理 V, 对给定的 f, 造一个其元如 $(x,\{f(x)\})$ 的新函数；对于公理 VI, 对给定的 x 研究其元都是形如 (u,u) 且 u 在 x 中的函数 [41].

定义 72 $x \times y = \{(u, v) : u \in x \wedge v \in y\}$.

```
Definition Cartesian x y : Class := \{\ λ u v, u ∈ x
  /\ v ∈ y \}\.

Notation "x × y" := (Cartesian x y)(at level 2, right
  associativity).

Hint Unfold Cartesian : set.
```

类 $x \times y$ 是 x 与 y 的 "**笛卡儿乘积**".

定理 73 u 是集 $\wedge$ y 是集 $\Longrightarrow \{u\} \times y$ 是集.

```
Lemma Ex_Lemma73 : ∀ u y,
  Ensemble u /\ Ensemble y -> ∃ f,Function f /\ dom(f) = y /\ ran(f) =
  [u] × y.
Proof.
  intros; destruct H.
  exists (\{\ λ w z, (w ∈ y /\ z = [u,w]) \}\).
  repeat split; intros.
  - red; intros; PP H1 a b; Ens.
  - destruct H1.
    apply AxiomII_P in H1; apply AxiomII_P in H2.
    destruct H1 as [_ [_ H1]]; destruct H2 as [_ [_ H2]].
    rewrite H2; auto.
  - apply AxiomI; split; intros.
    + apply AxiomII in H1; destruct H1 as [_ [t H1]].
      apply AxiomII_P in H1; tauto.
    + apply AxiomII; split; try Ens.
      exists [u,z]; apply AxiomII_P; split; auto.
      AssE z; apply Theorem49; split; auto.
      apply Theorem49; tauto.
  - apply AxiomI; split; intros.
    + apply AxiomII in H1; destruct H1, H1, H2.
      apply AxiomII_P in H2; destruct H2, H3.
      rewrite H4; apply AxiomII_P; repeat split; auto.
      * apply Theorem49; split; auto; AssE x0.
      * apply AxiomII; split; auto.
    + PP H1 a b; apply AxiomII_P in H2; destruct H2, H3.
      apply AxiomII; split; auto; exists b.
      apply AxiomII_P; repeat split; auto.
      * apply Theorem49; split; auto; AssE b.
      * apply Theorem19 in H; apply AxiomII in H3.
        destruct H3; rewrite H5; auto.
Qed.
```

```
Theorem Theorem73 : ∀ u y,
  Ensemble u /\ Ensemble y -> Ensemble ([u] × y).
Proof.
  intros; elim H; intros; apply Ex_Lemma73 in H; auto.
  destruct H,H,H2; rewrite <- H3; apply AxiomV; auto.
  rewrite H2; auto.
Qed.

Hint Resolve Theorem73 : set.
```

定理 74 x 是集 $\wedge$ y 是集 $\Longrightarrow x \times y$ 也是集.

```
Lemma Ex_Lemma74 : ∀ x y,
  Ensemble x /\ Ensemble y -> ∃ f, Function f /\ dom( f ) = x
  /\ ran( f ) = \{ λ z, (∃ u, u ∈ x /\ z = [u] × y) \}.
Proof.
  intros; destruct H.
  exists (\{\ λ u z, (u ∈ x /\ z = [u] × y) \}\).
  repeat split; intros.
  - red; intros; PP H1 a b; Ens.
  - destruct H1; apply AxiomII_P in H1; apply AxiomII_P in H2.
    destruct H1, H2, H3, H4; subst z; auto.
  - apply AxiomI; split; intros.
    + apply AxiomII in H1; destruct H1, H2.
      apply AxiomII_P in H2; tauto.
    + apply AxiomII; split; try AssE z.
      exists (([z]) × y); apply AxiomII_P.
      repeat split; auto; apply Theorem49; split; auto.
      apply Theorem73; auto.
  - apply AxiomI; split; intros.
    + apply AxiomII in H1; destruct H1, H2.
      apply AxiomII_P in H2; apply AxiomII.
      split; auto; exists x0; tauto.
    + apply AxiomII in H1; destruct H1, H2, H2.
      apply AxiomII; split; auto.
      exists x0; apply AxiomII_P; repeat split; auto.
      apply Theorem49; split; auto; AssE x0.
Qed.

Lemma Lemma74 : ∀ x y,
  Ensemble x /\ Ensemble y ->
  ⋃\{ λ z, (∃ u, u ∈ x /\ z = [u] × y) \} = x × y.
Proof.
  intros; apply AxiomI; split; intros.
```

```
  - apply AxiomII in H0; destruct H0, H1, H1.
    apply AxiomII in H2; destruct H2, H3, H3.
    rewrite H4 in H1; PP H1 a b.
    apply AxiomII_P in H5; destruct H5, H6.
    apply AxiomII_P; repeat split; auto.
    apply AxiomII in H6; destruct H6 as [_ H6].
    AssE x1; apply Theorem19 in H8.
    rewrite <- H6 in H3; auto.
  - PP H0 a b; apply AxiomII_P in H1; destruct H1, H2.
    apply AxiomII; split; auto.
    exists (([a]) × y); split; AssE a.
    + apply AxiomII_P; repeat split; auto.
      apply AxiomII; intros; auto.
    + apply AxiomII; split.
      * apply Theorem73; split; try apply H; auto.
      * exists a; split; auto.
Qed.

Theorem Theorem74 : ∀ x y,
  Ensemble x /\ Ensemble y -> Ensemble x × y.
Proof.
  intros; double H; double H0; destruct H0.
  apply Ex_Lemma74 in H; destruct H, H, H3.
  rewrite <- H3 in H0; apply AxiomV in H0; auto.
  rewrite H4 in H0; apply AxiomVI in H0.
  rewrite Lemma74 in H0; auto.
Qed.

Hint Resolve Theorem74 : set.
```

定理 75 f 是函数 $\wedge$ f 的定义域是集 $\Longrightarrow$ f 是集.

```
Theorem Theorem75 : forall f,
  Function f /\ Ensemble dom( f ) -> Ensemble f.
Proof.
  intros; destruct H.
  assert (Ensemble ran(f)); try apply AxiomV; auto.
  assert (Ensemble (dom( f)) × (ran( f))).
  { apply Theorem74; split; auto. }
  apply Theorem33 with (x:=(dom( f ) × ran( f ))); auto.
  unfold Included; intros; rewrite Theorem70 in H3; auto.
  PP H3 a b; rewrite <- Theorem70 in H4; auto; AssE [a,b].
  repeat split; auto; apply AxiomII_P; split; auto.
  generalize (Property_dom a b f H4); intro.
```

```
  generalize (Property_ran a b f H4); intro; tauto.
Qed.

Hint Resolve Theorem75 : set.
```

定义 76 $y^x = \{f : f\ 是函数 \wedge (f\ 的定义域 = x) \wedge (f\ 的值域 \subset y)\}$.

```
Definition Exponent y x : Class :=
  \{ λ f, (Function f /\ dom( f ) = x /\ (ran( f )) ⊂ y) \}.

Hint Unfold Exponent : set.
```

定理 77 x 是集 $\wedge$ y 是集 $\Longrightarrow y^x$ 也是集.

```
Theorem Theorem77 : ∀ x y,
  Ensemble x /\ Ensemble y -> Ensemble (Exponent y x).
Proof.
  intros; apply Theorem33 with (x:=(pow(x × y))).
  - apply Theorem38; auto; apply Theorem74; auto.
  - unfold Included; intros; apply Theorem38.
    + apply Theorem74; auto.
    + apply AxiomII in H0; destruct H0, H1, H2.
    unfold Included; intros; rewrite Theorem70 in H4; auto.
    PP H4 a b; rewrite <- Theorem70 in H5; auto.
    AssE [a,b]; apply AxiomII_P; split; auto.
    generalize (Property_dom a b z H5); intro; rewrite H2 in H7.
    generalize (Property_ran a b z H5); intro.
    unfold Included in H3; apply H3 in H8; split; auto.
Qed.

Hint Resolve Theorem77 : set.
```

定义 78 f 在 x 上 $\Longleftrightarrow f$ 是函数 $\wedge\ x = f$ 的定义域.

```
Definition On f x : Prop := (Function f /\ dom( f ) = x).

Hint Unfold On : set.
```

定义 79 f 到 $y \Longleftrightarrow f$ 是函数 $\wedge\ f$ 的值域 $\subset y$.

```
Definition To f y : Prop := (Function f /\ ran( f ) ⊂ y).

Hint Unfold To : set.
```

定义 80　f 到 y 上 $\iff f$ 是函数 $\wedge f$ 的值域 $= y$.

```
Definition Onto f y : Prop := (Function f /\ ran( f ) = y).

Hint Unfold Onto : set.

End A6.

Export A6.
```

3.7　良　　序

本节的许多结果在下面展开整数、序数与基数等理论中是不必要的. 而它们被包含在这里是因为自身是很有趣的, 并且这些方法是今后要用到的构造法的一种简化形式.

```
Require Export A_6.

(* A.7 良序 *)

Module A7.
```

定义 81　$xry \iff (x, y) \in r$.

```
Definition Rrelation x r y : Prop := [x,y] ∈ r.

Hint Unfold Rrelation : set.
```

如果 xry, 则 x 是 "r- **关系于** y", 或者 x 是 "r- **前于** y".

定义 82　r 连接 $x \iff \forall u \in x, \forall v \in x, urv \vee vru \vee u = v$.

```
Definition Connect r x : Prop :=
  ∀ u v, u ∈ x /\ v ∈ x -> (Rrelation u r v) \/ (Rrelation v r u) \/
  (u = v).

Hint Unfold Connect : set.
```

定义 83　r 在 x 中是传递的 $\iff (\forall u \in x, \forall v \in x, \forall w \in x, urv \wedge vrw \Longrightarrow urw)$.

```
Definition Transitive r x : Prop :=
  ∀ u v w, (u ∈ x /\ v ∈ x /\ w ∈ x /\ Rrelation u r v /\
  Rrelation v r w) -> Rrelation u r w.

Hint Unfold Transitive: set.
```

如果 x 在 r 中是 “**传递**”的, 则称 “r **序** x”, 如果 u 与 v 属于 x 并且 r 序 x, 特别有术语 “ur- **前于** v”.

定义 84 r 在 x 中是非对称的 $\iff (\forall u \in x, \forall v \in x, urv \implies \sim vru)$.

```
Definition Asymmetric r x : Prop :=
 ∀ u v, (u ∈ x /\ v ∈ x /\ Rrelation u r v) -> ~ Rrelation v r u.

Corollary Property_Asy : ∀ r x u, Asymmetric r x -> u ∈ x ->
 ~ Rrelation u r u.
Proof.
 intros; intro; unfold Asymmetric in H; eapply H; eauto.
Qed.

Hint Unfold Asymmetric: set. Hint Resolve Property_Asy: set.
```

x 在 r 中是“**是非对称**”的, 如果 u 与 v 属于 x 并且 ur- 前于 v, 则 v 不 r- 前于 u.

定义 85 (不等于) 之前已经使用, 在定义 15 之前已给出. 为与文献 [41] 一致, 仍给出下面的定义及其 Coq 形式化描述.

定义 85 (不等于) $x \neq y \iff \sim (x = y)$.

```
Definition Inequality (x y: Class) := ~ (x = y) .

Notation "x ≠ y" := (Inequality x y) (at level 70).
```

定义 86 z 是 x 的 r-首元 $\iff z \in x \wedge (\forall y \in x \implies \sim yrz)$.

```
Definition FirstMember z r x : Prop :=
 z ∈ x /\ (∀ y, y ∈ x -> ~ Rrelation y r z).

Hint Unfold FirstMember : set.
```

定义 87 r 良序 $x \iff r$ 连接 $x \wedge (\forall y \subset x \wedge y \neq 0 \implies \exists z,\ z$ 是 y 的 r-首元).

```
Definition WellOrdered r x : Prop :=
 Connect r x /\ (∀ y, y ⊂ x /\ y ≠ ∅ -> ∃ z, FirstMember z r y).

Hint Unfold WellOrdered : set.
```

定理 88 r 良序 $x \implies r$ 在 x 中是传递的 $\wedge\, r$ 在 x 中是非对称的.

```
Lemma Lemma88 : ∀ x u v w,
```

```
  Ensemble u -> Ensemble v -> Ensemble w ->
  x ∈ ([u] ∪ [v] ∪ [w]) -> x = u \/ x= v \/ x = w.
Proof.
  intros; apply Theorem19 in H; apply Theorem19 in H0.
  apply Theorem19 in H1; apply AxiomII in H2; destruct H2, H3.
  - left; apply AxiomII in H3; destruct H3; auto.
  - apply AxiomII in H3; destruct H3, H4.
    + right; left; apply AxiomII in H4; destruct H4; auto.
    + right; right; apply AxiomII in H4; destruct H4; auto.
Qed.

Theorem Theorem88 : ∀ r x, WellOrdered r x ->
  Transitive r x /\ Asymmetric r x .
Proof.
  intros; generalize H; intro.
  unfold WellOrdered in H0; destruct H0.
  assert (Asymmetric r x).
  { unfold Asymmetric; intros.
    destruct H2, H3; AssE u; AssE v.
    assert (([u | v] ⊂ x) /\ ([u | v] ≠ ∅)).
    { split.
      - unfold Included; intros; apply AxiomII in H7;
        destruct H7, H8.
        + apply AxiomII in H8; destruct H8; rewrite H9;
          try apply Theorem19; Ens.
        + apply AxiomII in H8; destruct H8; rewrite H9;
          try apply Theorem19; Ens.
      - apply Property_NotEmpty; exists u; apply AxiomII; split;
        auto.
        left; apply AxiomII; split; auto. }
    apply H1 in H7; destruct H7; unfold FirstMember in H7;
    destruct H7.
    apply Theorem46 in H7; auto; destruct H7; subst x0.
    - apply H8; apply AxiomII; split; auto; right;
      apply AxiomII; auto.
    - intro; apply H8 with u; auto; apply AxiomII; split;
      auto.
      left; apply AxiomII; split; auto. }
  split; auto; unfold Transitive; intros.
  - destruct H3, H4, H5, H6; unfold Connect in H0;
    specialize H0 with w u.
    destruct H0 as [H0 | [H0 | H0]]; try split; auto.
    + assert (([u] ∪ [v] ∪ [w] ⊂ x) /\ ([u] ∪ [v] ∪ [w] ≠ ∅)).
      { split.
        - unfold Included; intros; apply AxiomII in H8.
```

```
        destruct H8 as [_ H8]; destruct H8.
        + AssE u; apply Theorem19 in H9; apply AxiomII in H8.
          destruct H8; rewrite H10; auto.
        + apply AxiomII in H8; destruct H8 as [_ H8]; destruct H8.
          * AssE v; apply Theorem19 in H9; apply AxiomII in H8.
            destruct H8; rewrite H10; auto.
          * AssE w; apply Theorem19 in H9; apply AxiomII in H8.
            destruct H8; rewrite H10; auto.
      - intro; generalize (Theorem16 u); intro.
        apply H9; rewrite <- H8; apply AxiomII; split; Ens.
        left; apply AxiomII; split; intros; auto; Ens. }
    apply H1 in H8; destruct H8; unfold FirstMember in H8;
    destruct H8.
    assert (u ∈ ([u] ∪ [v] ∪ [w])).
    { apply Theorem4; left; apply AxiomII; split; Ens. }
    assert (v ∈ ([u] ∪ [v] ∪ [w])).
    { apply Theorem4; right; apply AxiomII; split; Ens.
      left; apply AxiomII; split; Ens. }
    assert (w ∈ ([u] ∪ [v] ∪ [w])).
    { apply Theorem4; right; apply AxiomII; split; Ens.
      right; apply AxiomII; split; Ens. }
    apply Lemma88 in H8; Ens; destruct H8 as [H8 | [H8|H8]];
    subst x0.
    * apply H9 in H12; contradiction.
    * apply H9 in H10; contradiction.
    * apply H9 in H11; contradiction.
  + subst w; unfold Asymmetric in H2;
  absurd (Rrelation u r v); auto.
Qed.

Hint Resolve Theorem88: set.
```

定义 89　y 是 x 的 r-截片 $\Longleftrightarrow y \subset x \wedge r$ 良序 $x \wedge (\forall u \in x, \forall v \in y, urv \Longrightarrow u \in y)$.

```
Definition Section y r x : Prop :=
  y ⊂ x /\ WellOrdered r x /\
  (∀ u v, (u ∈ x /\ v ∈ y /\ Rrelation u r v) -> u ∈ y).

Hint Unfold WellOrdered : set.
```

定义 89 是说, x 的一个子集 y 是 x 的一个 “r- **截片**”是指 r 良序 x, 同时没有 $x \sim y$ 的元 r-前于 y 的元.

定理 90　$n \neq 0 \wedge n$ 的每个元是 x 的 r- 截片 $\Longrightarrow \bigcup n$ 是 x 的 r- 截片 $\wedge \bigcap n$

是 x 的 r- 截片.

```
Theorem Theorem90 : ∀ n x r,
  n ≠ ∅ /\ (∀ y, y ∈ n -> Section y r x) ->
  Section (⋃n) r x /\ Section (⋂n) r x.
Proof.
  intros; destruct H; double H; apply Property_NotEmpty in H;
  destruct H; double H. apply H0 in H; red in H; destruct H, H3;
  split; unfold Section; intros.
  - split; try split; auto; intros.
    + unfold Included; intros; apply AxiomII in H5.
      destruct H5; apply H6 in H2; auto.
    + destruct H5, H6; apply AxiomII; split; intros; Ens.
      apply AxiomII in H6; destruct H6; double H8; apply H0 in H8.
      unfold Section in H8; eapply H8; split; eauto.
  - split; try split; auto; intros.
    + unfold Included; intros; apply AxiomII in H5;destruct H5, H6, H6.
      apply H0 in H7; unfold Section in H7;destruct H7 as [H7 _]; auto.
    + destruct H5, H6; apply AxiomII; split; intros; Ens.
      apply AxiomII in H6; destruct H6, H8, H8; double H9.
      apply H0 in H9; unfold Section in H9; destruct H9, H11.
      exists x1; split; auto; eapply H12; split; eauto.
Qed.

Hint Resolve Theorem90 : set.
```

定理 91 y 是 x 的 r-截片 $\wedge\, y \neq x \Longrightarrow \exists v \in x, y = \{u : u \in x \wedge urv\}$.

```
Theorem Theorem91 : ∀ x y r,
  Section y r x /\ y≠x ->
  (∃ v, v ∈ x /\ y = \{ λ u, u ∈ x /\ Rrelation u r v \}).
Proof.
  intros; destruct H.
  assert (exists v0, FirstMember v0 r (x ~ y)).
  { unfold Section in H; destruct H, H1; unfold WellOrdered in H1;
    destruct H1.
    assert ((x ~ y) ⊂ x).
    { unfold Included; intros; apply AxiomII in H4; tauto. }
    generalize (classic (x ~ y = ∅)); intro; destruct H5.
    - apply Property_∅ in H; apply H in H5.
      apply Property_Ineq in H0; contradiction.
    - apply H3; split; auto. }
  destruct H1; unfold FirstMember in H1; destruct H1; exists x0.
  apply AxiomII in H1; destruct H1, H3; split; auto; apply AxiomI.
  split; intros; unfold Section in H; destruct H, H6.
```

```
  - apply AxiomII; repeat split; Ens; assert (z ∈ x); auto.
    unfold WellOrdered in H6; destruct H6 as [H6 _];
    unfold Connect in H6.
    specialize H6 with x0 z; destruct H6 as [H6 | [H6 | H6]]; auto.
    + assert (x0 ∈ y). { apply H7 with z; repeat split; auto. }
      apply AxiomII in H4; destruct H4; contradiction.
    + apply AxiomII in H4; destruct H4; subst x0; contradiction.
  - apply AxiomII in H5; destruct H5, H8.
    generalize (classic (z ∈ (x ~ y))); intro; destruct H10.
    + apply H2 in H10; contradiction.
    + generalize (classic (z ∈ y)); intro; destruct H11; auto.
      elim H10; apply AxiomII; repeat split; auto;apply AxiomII; tauto.
Qed.

Hint Resolve Theorem91 : set.
```

定理 92　x 和 y 是 z 的 r- 截片 $\Longrightarrow x \subset y \vee y \subset x$.

```
Theorem Theorem92 : ∀ x y z r,
  Section x r z /\ Section y r z -> x ⊂ y \/ y ⊂ x.
Proof.
  intros; destruct H.
  generalize (classic (x = z)); intro; destruct H1.
  - right; red in H0; subst z; tauto.
  - generalize (classic (y = z)); intro; destruct H2.
    + left; red in H; subst z; tauto.
    + apply Lemma_y with (A:=(Section x r z)) in H1; auto.
      apply Lemma_y with (A:=(Section y r z)) in H2; auto.
      apply Theorem91 in H1; destruct H1, H1.
      apply Theorem91 in H2; destruct H2, H2.
      unfold Section in H; destruct H as [_ [H _]].
      unfold WellOrdered in H; destruct H as [H _].
      unfold Section in H0; destruct H0, H5.
      apply Theorem88 in H5; destruct H5; unfold Transitive in H5.
      assert ((x0 ∈ z) /\ (x1 ∈ z)); try split; auto.
      unfold Connect in H; generalize (H _ _ H8); intros.
      destruct H9 as [H9 | [H9 | H9]].
      * left; unfold Included; intros; rewrite H3 in H10.
        apply AxiomII in H10; destruct H10, H11;
        rewrite H4; apply AxiomII.
        repeat split; auto; apply H5 with x0; auto.
      * right; unfold Included; intros; rewrite H4 in H10.
        apply AxiomII in H10; destruct H10, H11; rewrite H3;
        apply AxiomII.
        repeat split; auto; apply H5 with x1; auto.
```

```
    * right; subst x0; rewrite H3, H4; unfold Included; intros; auto.
Qed.

Hint Resolve Theorem92 : set.
```

定义 93 f 是 r-s 保序的 $\Longleftrightarrow$ f 是函数, r 良序 f 的定义域, s 良序 f 的值域 $\wedge$ ($\forall u \in f$ 的定义域, $\forall v \in f$ 的定义域, $urv \Longrightarrow f(u)sf(v)$).

```
Definition Order_Pr f r s : Prop :=
  Function f /\ WellOrdered r dom(f) /\ WellOrdered s ran(f) /\
  (∀ u v, u ∈ dom(f) /\ v ∈ dom(f) /\ Rrelation u r v ->
   Rrelation f[u] s f[v]).

Hint Unfold Order_Pr : set.
```

定理 94 x 为 y 的 r- 截片 $\wedge$ f 是在 x 上到 y 的 r-r 保序函数 $\Longrightarrow$ $\forall u \in x$, $\sim f(u)ru$.

```
Theorem Theorem94 : ∀ x r y f,
  Section x r y /\ Order_Pr f r r /\ On f x /\ To f y ->
  (∀ u, u ∈ x -> ~ Rrelation f[u] r u).
Proof.
  intros; destruct H, H1, H2.
  unfold Order_Pr in H1; destruct H1, H4, H5.
  unfold On in H2; destruct H2 as [H2 H7].
  unfold To in H3; destruct H3 as [_ H3].
  generalize (classic (\{ λ u, u ∈ x /\ Rrelation f[u] r u \} = ∅)).
  intros; destruct H8.
  - intro; assert (u ∈ ∅).
    { rewrite <- H8; apply AxiomII; repeat split; Ens. }
    generalize (Theorem16 u); intro; contradiction.
  - unfold Section in H; destruct H, H9.
    assert (\{ λ u, u ∈ x /\ Rrelation f[u] r u \} ⊂ y).
    { unfold Included; intros; apply AxiomII in H11;
    destruct H11, H12; auto. }
    unfold WellOrdered in H9; destruct H9.
    add (\{ λ u, u ∈ x /\ Rrelation f[u] r u \} ≠ ∅) H11.
    apply H12 in H11; destruct H11; unfold FirstMember in H11;
    destruct H11.
    apply AxiomII in H11; destruct H11, H14.
    assert (f[x0] ∈ ran( f)).
    { rewrite <- H7 in H14; apply Property_Value in H14; auto.
      apply Property_ran in H14; auto. }
    assert (f[x0] ∈ y); auto; subst x.
    assert (f[x0] ∈ \{ λ u, u ∈ dom(f) /\ Rrelation f[u] r u \}).
```

```
    { apply AxiomII; repeat split; try Ens.
      apply H6; repeat split; auto; apply H10 with x0; split; auto. }
    apply H13 in H7; contradiction.
Qed.

Hint Resolve Theorem94 : set.
```

于是 r-r 保序函数在一个 r- 截片上不能把它定义域的元映成一个 r-前趋.

像定理 94 这样的证明是依据使定理不成立的首元的研究. 这种证明叫 “**归纳法**”证明.

定义 95　f 是 1-1 函数 $\Longleftrightarrow f$ 与 f^{-1} 都是函数.

```
Definition Function1_1 f : Prop := Function f /\ Function (f⁻¹).

Hint Unfold Function1_1 : set.
```

定理 96　f 是 r-s 保序的 $\Longrightarrow f$ 是 1-1 函数 $\wedge f^{-1}$ 是 s-r 保序的.

```
Lemma Lemma96 : ∀ f, dom( f ) = ran( f⁻¹ ).
Proof.
  intros; apply AxiomI; split; intros.
  - apply AxiomII in H; destruct H, H0; apply AxiomII; split; auto.
    exists x; apply AxiomII_P; split; auto; apply Lemma61; Ens.
  - apply AxiomII in H; destruct H, H0; apply AxiomII.
    split; auto; exists x; apply AxiomII_P in H0; tauto.
Qed.

Lemma Lemma96' : ∀ f, ran( f ) = dom( f⁻¹ ).
Proof.
  intros; apply AxiomI; split; intros.
  - apply AxiomII in H; destruct H, H0; apply AxiomII; split; auto.
    exists x; apply AxiomII_P; split; auto; apply Lemma61; Ens.
  - apply AxiomII in H; destruct H, H0; apply AxiomII.
    split; auto; exists x; apply AxiomII_P in H0; tauto.
Qed.

Lemma Lemma96'' : ∀ f u,
  Function f -> Function f⁻¹ -> u ∈ ran(f) -> (f⁻¹)[u] ∈ dom(f).
Proof.
  intros; rewrite Lemma96' in H1; apply Property_Value in H1; auto.
  apply AxiomII_P in H1; destruct H1; apply Property_dom in H2; auto.
Qed.

Lemma Lemma96''' : ∀ f u,
```

```
  Function f -> Function f⁻¹ -> u ∈ ran(f) -> u = f [ (f⁻¹)[u] ].
Proof.
  intros; generalize (Lemma96'' _ _ H H0 H1); intro.
  apply Property_Value in H2; auto; rewrite Lemma96' in H1.
  apply Property_Value in H1; auto; apply AxiomII_P in H1; destruct H1.
  red in H; destruct H; eapply H4; eauto.
Qed.

Theorem Theorem96 : ∀ f r s,
  Order_Pr f r s -> Function1_1 f /\ Order_Pr (f⁻¹) s r.
Proof.
  intros; unfold Order_Pr in H; destruct H, H0, H1.
  assert (Function1_1 f).
  { unfold Function1_1; split; auto; unfold Function; split; intros.
    - red; intros; PP H3 a b; Ens.
    - destruct H3; rename y into u; rename z into v.
      apply AxiomII_P in H3; destruct H3; apply AxiomII_P in H4;
      destruct H4.
      double H5; double H6; apply Property_dom in H5;
      apply Property_dom in H6.
      double H7; double H8; apply Property_dom in H7;
      apply Property_dom in H8.
      rewrite Theorem70 in H9; auto; apply AxiomII_P in H9;
      destruct H9 as [_ H9].
      rewrite Theorem70 in H10; auto; apply AxiomII_P in H10;
      destruct H10 as [_ H10].
      rewrite H10 in H9; symmetry in H9; clear H10.
      apply Property_Value in H7; apply Property_Value in H8; auto.
      apply Property_ran in H7; apply Property_ran in H8.
      double H0; double H1; apply Theorem88 in H11; destruct H11.
      unfold WellOrdered in H1; destruct H1 as [H1 _].
      unfold Connect in H1; specialize H1 with f[u] f[v].
      unfold WellOrdered in H0; destruct H0.
      unfold Connect in H0; specialize H0 with u v.
      destruct H0 as [H0 | [H0 | H0]]; try split; auto.
      + assert (Rrelation f[u] s f[v]); try apply H2; try tauto.
        rewrite H9 in H14; generalize (Property_Asy _ _ _ H12 H8).
        intro; contradiction.
      + assert (Rrelation f[v] s f[u]); try apply H2; try tauto.
        rewrite H9 in H14; generalize (Property_Asy _ _ _ H12 H8).
        intro; contradiction. }
  split; auto; destruct H3 as [_ H3]; unfold Order_Pr; intros.
  repeat rewrite <- Lemma96; repeat rewrite <- Lemma96'; split; auto.
  split; auto; split; intros; auto; destruct H4, H5.
  assert ((f⁻¹)[u] ∈ dom(f)); try apply Lemma96''; auto.
```

```
  assert ((f⁻¹)[v] ∈ dom(f)); try apply Lemma96''; auto.
  unfold WellOrdered in H0; destruct H0 as [H0_]; unfold Connect in H0.
  specialize H0 with (f⁻¹)[u] (f⁻¹)[v].
  destruct H0 as [H0 | [H0 | H0]]; try split; auto.
  - assert (Rrelation f[ (f⁻¹)[v] ] s f[ (f⁻¹)[u] ]); auto.
    rewrite <- Lemma96''' in H9; rewrite <- Lemma96''' in H9; auto.
    apply Theorem88 in H1; destruct H1; unfold Asymmetric in H10.
    generalize (Lemma_y _ _ H5 (Lemma_y _ _ H4 H9)); intro.
    generalize (H10 _ _ H11); intro; contradiction.
  - assert (f[ (f⁻¹)[u] ] = f[ (f⁻¹)[v] ]); rewrite H0; auto.
    rewrite <- Lemma96''' in H9; rewrite <- Lemma96''' in H9; auto.
    apply Theorem88 in H1; destruct H1; rewrite H9 in H6.
    apply Property_Asy with (r:= s) in H5; tauto.
Qed.

Hint Resolve Theorem96 : set.
```

定理 97　f 与 g 是 r-s 保序的, f 的定义域与 g 的定义域均为 x 的 r- 截片, f 的值域与 g 的值域均为 y 的 s-截片 $\Longrightarrow f\subset g\vee g\subset f$.

```
Lemma Lemma97 : ∀ y r x,
  WellOrdered r x -> y ⊂ x -> WellOrdered r y.
Proof.
  intros; unfold WellOrdered in H; destruct H.
  unfold WellOrdered; intros; split; intros.
  - red; intros; apply H; destruct H2; split; auto.
  - specialize H1 with y0; apply H1; destruct H2.
    split; auto; eapply Theorem28; eauto.
Qed.

Lemma Lemma97' : ∀ f g u r s v x y,
  Order_Pr f r s /\ Order_Pr g r s ->
  FirstMember u r (\{ λ a ,a ∈ (dom(f) ∩ dom(g)) /\ f [a] ≠ g [a] \})
  -> g[v] ∈ ran(g) -> Section ran(f) s y -> Section dom(f) r x ->
  Section dom(g) r x -> Rrelation g [v] s g [u] ->
  f[u] = g[v] -> f ⊂ g \/ g ⊂ f.
Proof.
  intros.
  unfold FirstMember in H0; destruct H0.
  apply AxiomII in H0; destruct H0, H8.
  apply AxiomII in H8; destruct H8 as [_ [H8 H10]].
  destruct H; unfold Order_Pr in H, H11.
  apply Property_Value in H8; apply Property_Value in H10; try tauto.
  apply Property_ran in H8; apply Property_ran in H10; auto.
  assert (Rrelation v r u).
```

```
  { elim H11; intros; clear H13.
    apply Theorem96 in H11; destruct H11 as [_ H11].
    red in H11; destruct H11 as [H11 [_ [_ H13]]].
    double H1; double H10; rewrite Lemma96' in H14, H15.
    apply Property_Value' in H10; auto; apply Property_dom in H10.
    rewrite Lemma96 in H10; apply Property_Value' in H1; auto.
    apply Property_dom in H1; rewrite Lemma96 in H1.
    rewrite Lemma96''' with (f:=g⁻¹);
    try (rewrite Theorem61; apply H12); auto.
    pattern v; rewrite Lemma96''' with (f:=(g⁻¹));
    try rewrite Theorem61; try apply H11; try apply H12; auto. }
  assert (v ∈ \{ λ a, a ∈ (dom(f) ∩ dom(g)) /\ f [a] ≠ g [a] \}).
  { apply Property_Value' in H1; try tauto; apply Property_dom in H1.
    apply Property_Value' in H8; try tauto; apply Property_dom in H8.
    apply AxiomII; repeat split; try Ens.
    - apply AxiomII; repeat split; try Ens.
      apply H3 with u; repeat split; auto.
      unfold Section in H4; apply H4; auto.
    - intro.
      assert (v ∈ dom(f)).
      { apply H3 with u; repeat split; auto; apply H4 in H1; auto. }
      assert (Rrelation f [v] s f [u]).
      { apply H; repeat split; auto. }
      rewrite H13 in H15; unfold Section in H2; destruct H2, H16.
      generalize (Lemma97 _ _ _ H16 H2); intro.
      apply Theorem88 in H18; destruct H18.
      rewrite <- H13 in H15; rewrite H6 in H15; rewrite <- H13 in H15.
      apply Property_Value in H14; try tauto;
      apply Property_ran in H14.
      generalize (Property_Asy _ _ _ H19 H14);
      intro; contradiction. }
  apply H7 in H13; contradiction.
Qed.

Lemma Lemma97'' : ∀ f g,
  \{ λ a, a ∈ (dom(f) ∩ dom(g)) /\ f[a] ≠ g[a] \} =
  \{ λ a, a ∈ (dom(g) ∩ dom(f)) /\ g[a] ≠ f[a] \}.
Proof.
  intros; apply AxiomI; split; intros; rewrite Theorem6';
  apply AxiomII in H;
  apply AxiomII; repeat split; try tauto; apply Property_Ineq; tauto.
Qed.

Lemma Lemma97''' : ∀ f g, f ⊂ g \/ g ⊂ f <-> g ⊂ f \/ f ⊂ g.
Proof.
```

```
  intros; split; intros; destruct H; tauto.
Qed.

Theorem Theorem97 : ∀ f g r s x y,
  Order_Pr f r s /\ Order_Pr g r s ->
  Section dom(f) r x /\ Section dom(g) r x ->
  Section ran(f) s y /\ Section ran(g) s y -> f ⊂ g \/ g ⊂ f.
Proof.
  intros; destruct H, H0, H1.
  assert (Order_Pr (g⁻¹) s r); { apply Theorem96 in H2; tauto. }
  generalize (classic (\{λ a, a ∈ (dom(f) ∩ dom(g)) /\ f[a] ≠ g[a] \}
  = ∅)).
  intro; destruct H6.
  - generalize (Lemma_y _ _ H0 H3); intro.
    unfold Order_Pr in H;destruct H;unfold Order_Pr in H2;destruct H2.
    generalize (Theorem92 _ _ _ _ H7); intro; destruct H10.
    + left; unfold Included; intros.
      rewrite Theorem70 in H11; auto; PP H11 a b; double H12.
      rewrite <- Theorem70 in H12; auto; apply Property_dom in H12.
      apply AxiomII_P in H13; destruct H13.
      rewrite Theorem70; auto;apply AxiomII_P;split; auto; rewrite H14.
      generalize (classic (f[a] = g[a])); intro; destruct H15; auto.
      assert (a ∈ \{ λ a, a ∈ (dom(f) ∩ dom(g)) /\ f[a] ≠ g[a] \}).
      { apply AxiomII; split; Ens; split; auto.
        apply Theorem30 in H10; rewrite H10; auto. }
      eapply AxiomI in H6; apply H6 in H16.
      generalize (Theorem16 a); contradiction.
    + right; unfold Included; intros.
      rewrite Theorem70 in H11; auto; PP H11 a b; double H12.
      rewrite <- Theorem70 in H12; auto; apply Property_dom in H12.
      apply AxiomII_P in H13; destruct H13.
      rewrite Theorem70; auto;apply AxiomII_P;split; auto; rewrite H14.
      generalize (classic (f[a] = g[a])); intro; destruct H15; auto.
      assert (a ∈ \{ λ a, a ∈ (dom(f) ∩ dom(g)) /\ f[a] ≠ g[a] \}).
      { apply AxiomII; split; Ens; split; auto. apply Theorem30 in H10.
        rewrite Theorem6' in H10; rewrite H10; auto. }
      eapply AxiomI in H6; apply H6 in H16.
      generalize (Theorem16 a); contradiction.
  - assert (\{ λ a, a ∈ (dom(f) ∩ dom(g)) /\ f[a] ≠ g[a] \}⊂dom(f)).
    { unfold Included; intros; apply AxiomII in H7; destruct H7, H8.
      apply Theorem4' in H8; tauto. }
    double H2; double H; unfold Order_Pr in H9; destruct H9, H10, H11.
    unfold WellOrdered in H10; destruct H10.
    generalize (Lemma_y _ _ H7 H6); intro.
    apply H13 in H14; destruct H14 as [u H14].
```

```
double H14; unfold FirstMember in H15; destruct H15.
apply AxiomII in H15; destruct H15, H17; unfold Order_Pr in H2.
destruct H2 as [H19 [_ [H2 _]]]; apply AxiomII in H17.
destruct H17 as [_ [H17 H20]]; double H17; double H20.
apply Property_Value in H17; apply Property_Value in H20; auto.
apply Property_ran in H17; apply Property_ran in H20.
generalize (Lemma_y _ _ H1 H4); intro.
apply Theorem92 in H23; auto; destruct H23.
+ apply H23 in H17; double H17.
  apply AxiomII in H17; destruct H17 as [_ [v H17]].
  rewrite Theorem70 in H17; auto; apply AxiomII_P in H17.
  destruct H17; rewrite H25 in H24.
  generalize (Lemma_y _ _ H24 H20); intro.
  unfold WellOrdered in H2; destruct H2 as [H2 _].
  unfold Connect in H2; apply H2 in H26.
  destruct H26 as [H26 | [H26 | H26]].
  * apply (Lemma97' f g u r s v x y); auto.
  * rewrite <- H25 in H26.
    assert (g [u] ∈ ran( f)).
    { unfold Section in H4; apply H4 in H20.
      unfold Section in H1; apply H1 with f[u]; repeat split; auto.
      apply Property_ran with u; apply Property_Value; auto. }
    double H27; apply AxiomII in H27; destruct H27 as [_ [v1 H27]].
    rewrite Theorem70 in H27; auto.
    apply AxiomII_P in H27; destruct H27 as [_H27].
    rewrite H27 in H26, H28; rewrite Lemma97'' in H14;
    apply Lemma97'''.
    apply (Lemma97' g f u r s v1 x y); try tauto.
  * rewrite H26 in H25; contradiction.
+ apply H23 in H20; double H20.
  apply AxiomII in H20; destruct H20 as [_ [v H20]].
  rewrite Theorem70 in H20; auto.
  apply AxiomII_P in H20; destruct H20; rewrite H25 in H24.
  generalize (Lemma_y _ _ H17 H24); intro.
  unfold WellOrdered in H11; destruct H11 as [H11 _].
  unfold Connect in H11; apply H11 in H26.
  destruct H26 as [H26 | [H26 | H26]]; try contradiction.
  * rewrite <- H25 in H26.
    assert (f[u] ∈ ran( g)).
    { unfold Section in H1; apply H1 in H17.
      unfold Section in H4; eapply H4; repeat split; eauto.
      apply Property_ran with u; apply Property_Value; auto. }
    double H27; apply AxiomII in H27; destruct H27 as [_ [v1 H27]].
    rewrite Theorem70 in H27; auto.
    apply AxiomII_P in H27; destruct H27 as [_H27].
```

```
      rewrite H27 in H26, H28.
      apply (Lemma97' f g u r s v1 x y); try tauto.
    * rewrite Lemma97'' in H14; apply Lemma97'''.
      apply (Lemma97' g f u r s v x y); try tauto.
    * rewrite <- H25 in H26; contradiction.
Qed.

Hint Resolve Theorem97 : set.
```

定义 98　f 在 x 和 y 中 r-s 保序 $\Longleftrightarrow$ r 良序 x, s 良序 y, f 是 r-s 保序, f 的定义域是 x 的 r-截片, f 的值域是 y 的 s-截片.

```
Definition Order_PXY f x y r s : Prop :=
  WellOrdered r x /\ WellOrdered s y /\ Order_Pr f r s /\
  Section dom(f) r x /\ Section ran(f) s y.

Hint Unfold Order_PXY : set.
```

按照定理 97, 如果 f 和 g 在 x 和 y 中都是 r-s 保序的, 则 $f \subset g$ 或者 $g \subset f$.

定理 99　r 良序 x, s 良序 y $\Longrightarrow$ $\exists f$, f 是函数, f 在 x 和 y 中 r-s 保序, (f 的定义域 $= x$) $\vee$ (f 的值域 $= y$).

```
Lemma Lemma99 : ∀ y r x,
  WellOrdered r x -> Section y r x -> WellOrdered r y.
Proof.
  intros; red in H0; eapply Lemma97; eauto; tauto.
Qed.

Lemma Lemma99' : ∀ a b f z,
  ~ a ∈ dom(f) -> Ensemble a -> Ensemble b ->
  (z ∈ dom(f) -> (f ∪ [[a,b]]) [z] = f [z]).
Proof.
  intros;apply AxiomI;split;intros;apply AxiomII in H3;destruct H3;
  apply AxiomII; split; intros; auto.
  - apply H4; apply AxiomII in H5; destruct H5;
    apply AxiomII; split; auto.
    apply AxiomII; split; Ens.
  - apply H4; apply AxiomII in H5; destruct H5.
    apply AxiomII in H6; destruct H6, H7; apply AxiomII; auto.
    assert ([a, b] ∈ 𝒰). {apply Theorem19; apply Theorem49; tauto.}
    apply AxiomII in H7; destruct H7; apply H9 in H8.
    apply Theorem55 in H8; destruct H8; Ens.
    rewrite H8 in H2; contradiction.
Qed.
```

```
Lemma Lemma99'' : ∀ a b f z,
   ~ a ∈ dom(f) -> Ensemble a -> Ensemble b ->
  (z = a -> (f ∪ [[a,b]]) [z] = b).
Proof.
  intros; apply AxiomI; split; intros; subst z.
  - apply AxiomII in H3; destruct H3; apply H3;
    apply AxiomII; split;
    auto.
    apply AxiomII; split; try apply Theorem49; try tauto.
    right; apply AxiomII; split; try apply Theorem49; try tauto.
  - apply AxiomII; split; intros; Ens.
    apply AxiomII in H2; destruct H2;
    apply AxiomII in H4; destruct H4, H5.
    + apply Property_dom in H5; contradiction.
    + apply AxiomII in H5; destruct H5.
      assert ([a, b] ∈ 𝓤). {apply Theorem19; apply Theorem49; tauto.}
      generalize (H6 H7); intro; apply Theorem55 in H8;
      apply Theorem49 in H4; auto; destruct H8; rewrite H9; auto.
Qed.

Lemma Lemma99''' : ∀ r x a b,
  Section y r x -> a ∈ y -> ~ b ∈ y -> b ∈ x -> Rrelation a r b.
Proof.
  intros; unfold Section in H; destruct H, H3.
  unfold WellOrdered in H3; destruct H3; unfold Connect in H3.
  assert (a ∈ x); auto.
  generalize (Lemma_y _ _ H2 H6); intro; apply H3 in H7.
  destruct H7 as [H7 | [H7 | H7]]; auto.
  - assert (b ∈ y). { eapply H4; eauto. }
    contradiction.
  - rewrite H7 in H1; contradiction.
Qed.

Definition En_f x y r s :=
  \{\ λ u v, u ∈ x /\ (exists g, Function g /\ Order_PXY g x y r s /\
  u ∈ dom(g) /\ [u,v] ∈ g ) \}\.

Theorem Theorem99 : ∀ r s x y,
  WellOrdered r x /\ WellOrdered s y ->
  ∃ f, Function f /\ Order_PXY f x y r s /\ ((dom(f) = x) \/
  (ran(f) = y)).
Proof.
  intros.
  assert (Function (En_f x y r s)).
```

```
{ unfold Function; split; intros.
   - unfold Relation; intros; PP H0 a b; eauto.
   - destruct H0.
     apply AxiomII_P in H0; destruct H0, H2, H3, H3, H4, H5.
     unfold Order_PXY in H4; destruct H4 as [_ [_ [H4 [H7 H8]]]].
     apply AxiomII_P in H1; destruct H1, H9, H10, H10, H11, H12.
     unfold Order_PXY in H11; destruct H11 as [_[_[H11[H14 H15]]]].
     assert (x1 ⊂ x2 \/ x2 ⊂ x1).
     { apply (Theorem97 x1 x2 r s x y); tauto. }
     destruct H16.
     * apply H16 in H6; eapply H10; eauto.
     * apply H16 in H13; eapply H3; eauto. }
exists (En_f x y r s); split; auto.
assert (Section (dom(En_f x y r s)) r x).
{ unfold Section; split.
  - unfold Included; intros; apply AxiomII in H1; destruct H1, H2.
    apply AxiomII_P in H2; tauto.
  - split; try tauto; intros; destruct H1, H2.
    apply AxiomII in H2; destruct H2, H4.
    apply AxiomII_P in H4; destruct H4, H5, H6;
    apply AxiomII; split; Ens.
    exists ((En_f x y r s)[u]); apply Property_Value; auto.
    apply AxiomII; split; Ens.
    assert (u ∈ dom( x1)).
    { destruct H6, H7; unfold Order_PXY in H7;
      destruct H7, H9, H10, H11.
      unfold Section in H11; destruct H11, H13; apply H14 with v.
      destruct H8; tauto. }
    exists (x1[u]).
    apply AxiomII_P; repeat split; auto.
    + apply Theorem49; split; Ens.
      apply Theorem19; apply Theorem69; try tauto.
    + exists x1; split; try tauto; split; try tauto; split; auto.
      apply Property_Value; try tauto. }
assert (Section (ran(En_f x y r s)) s y).
{ unfold Section; split.
  - unfold Included; intros; apply AxiomII in H2; destruct H2, H3.
    apply AxiomII_P in H3; destruct H3, H4, H5, H5, H6, H7.
    unfold Order_PXY in H6; destruct H6 as [_ [_ [_ [_ H6]]]].
    unfold Section in H6; destruct H6 as [H6 _].
    apply Property_ran in H8; auto.
  - split; try tauto; intros; destruct H2, H3.
    apply AxiomII in H3; destruct H3, H5.
    apply AxiomII_P in H5; destruct H5, H6, H7.
    apply AxiomII; split; Ens.
```

```
    exists (x1⁻¹[u]); apply AxiomII_P;
    destruct H7 as [H7 [H8 [H9 H10]]].
    double H8; unfold Order_PXY in H8;
    destruct H8 as [_ [_ [H12 [H13 H8]]]].
    generalize H11 as H20; intro.
    unfold Order_PXY in H11; destruct H11 as [H11 [_ H19]].
    unfold Section in H8; destruct H8 as [H8 [_ H15]].
    assert (u ∈ ran( x1)).
    { apply Property_ran in H10; apply H15 with v; tauto. }
    generalize H14 as H21; intro.
    apply Theorem96 in H12; destruct H12.
    unfold Function1_1 in H12; destruct H12.
    apply Lemma96'' in H14; auto.
    repeat split; auto.
    * apply Theorem49; split; Ens.
    * apply Property_Value in H14; auto.
      rewrite <- Lemma96''' in H14; auto.
      apply Property_dom in H14; destruct H19 as[_[[H19 _] _]]; auto.
    * exists x1; split; try tauto. split; try tauto; split; auto.
      apply Property_Value in H14; auto.
      rewrite <- Lemma96''' in H14; auto. }
assert (Order_PXY (En_f x y r s) x y r s).
{ unfold Order_PXY; split; try tauto; split; try tauto.
  split; [idtac | tauto].
  unfold Order_Pr; split; auto; destruct H; split;
  try eapply Lemma99; eauto.
  split; intros; try eapply Lemma99; eauto.
  destruct H4, H5; double H4; double H5.
  apply Property_Value in H4; apply Property_Value in H5; auto.
  apply AxiomII_P in H4; apply AxiomII_P in H5.
  destruct H4 as [H4 [H9 [g1 [H10 [H11 [H12 H13]]]]]].
  destruct H5 as [H5 [H14 [g2 [H15 [H16 [H17 H18]]]]]].
  rewrite Theorem70 in H13; rewrite Theorem70 in H18; auto.
  apply AxiomII_P in H13; destruct H13 as [_ H13].
  apply AxiomII_P in H18; destruct H18 as [_ H18].
  rewrite H13, H18; clear H13 H18.
  unfold Order_PXY in H11; destruct H11 as [_ [_ [H11 [H13 H18]]]].
  unfold Order_PXY in H16; destruct H16 as [_ [_ [H16 [H19 H20]]]].
  generalize (Lemma_y _ _ H11 H16); intro.
  apply (Theorem97 g1 g2 r s x y) in H21; auto.
  apply Property_Value in H12; apply Property_Value in H17; auto.
  destruct H21.
  - apply H21 in H12; double H12; rewrite Theorem70 in H12; auto.
    apply AxiomII_P in H12; destruct H12 as [_ H12]; rewrite H12.
    apply Property_dom in H22; apply Property_dom in H17;
```

```
    apply H16; tauto.
  - apply H21 in H17; double H17; rewrite Theorem70 in H17; auto.
    apply AxiomII_P in H17; destruct H17 as [_ H17]; rewrite H17.
    apply Property_dom in H12; apply Property_dom in H22;
    apply H11; tauto. }
split; auto.
apply NNPP; intro; apply not_or_and in H4; destruct H4.
assert (exists u, FirstMember u r (x ~ dom( En_f x y r s))).
{ unfold Section in H1; destruct H1, H6.
  assert ((x ~ dom( En_f x y r s)) ⊂ x).
  { red; intros; apply AxiomII in H8; tauto. }
  assert ((x ~ dom( En_f x y r s)) <> ∅).
  { intro; apply Property_∅ in H1; apply H1 in H9; apply H4; auto. }
  generalize (Lemma97 _ _ _ H6 H8); intro.
  apply H10; repeat split; auto; red; auto. }
assert (exists v, FirstMember v s (y ~ ran( En_f x y r s))).
{ unfold Section in H2; destruct H2, H7.
  assert ((y ~ ran( En_f x y r s)) ⊂ y).
  { red; intros; apply AxiomII in H9; tauto. }
  assert ((y ~ ran( En_f x y r s)) <> ∅).
  { intro; apply Property_∅ in H2; apply H2 in H10; apply H5; auto. }
  generalize (Lemma97 _ _ _ H7 H9); intro.
  apply H11; repeat split; auto; red; auto. }
destruct H6 as [u H6]; destruct H7 as [v H7].
unfold FirstMember in H6; unfold FirstMember in H7; destruct H6, H7.
apply AxiomII in H6; destruct H6 as [_ [H6 H10]].
apply AxiomII in H10; destruct H10 as [_ H10].
apply H10; apply AxiomII; split; Ens.
exists v; apply AxiomII_P.
split; try apply Theorem49; split; try Ens.
exists ((En_f x y r s) ∪ [[u,v]]).
assert (Function (En_f x y r s ∪ [[u, v]])).
{ assert ([u, v] ∈ 𝒰) as H18.
  { apply Theorem19; apply Theorem49; split; try Ens. }
  unfold Function; split; intros.
  - unfold Relation; intros.
    apply AxiomII in H11; destruct H11 as [H11 [H12 | H12]].
    * PP H12 a b; eauto.
    * apply AxiomII in H12; exists u,v; apply H12; auto.
  - destruct H11; apply AxiomII in H11; apply AxiomII in H12.
    destruct H11 as [H11 [H13 | H13]], H12 as [H12 [H14 | H14]].
    * unfold Function in H0; eapply H0; eauto.
    * apply Property_dom in H13; apply AxiomII in H14; destruct H14.
       apply Theorem55 in H15; apply Theorem49 in H12; auto.
       destruct H15; rewrite H15 in H13; contradiction.
```

```
    * apply Property_dom in H14; apply AxiomII in H13; destruct H13.
      apply Theorem55 in H15; apply Theorem49 in H11; auto.
      destruct H15; rewrite H15 in H14; contradiction.
    * apply AxiomII in H13; destruct H13; apply Theorem55 in H15;
      apply Theorem49 in H13; auto.
      apply AxiomII in H14; destruct H14; apply Theorem55 in H16;
      apply Theorem49 in H12; auto.
      destruct H15, H16; rewrite H17; auto. }
split; auto.
assert (Section (dom(En_f x y r s ∪ [[u, v]])) r x).
{ unfold Section; split.
  - unfold Included; intros; apply AxiomII in H12; destruct H12, H13.
    apply AxiomII in H13; destruct H13, H14.
    * apply Property_dom in H14;unfold Section in H1;apply H1;auto.
    * apply AxiomII in H14; destruct H14.
      assert ([u, v] ∈ 𝓤).
      { apply Theorem19; apply Theorem49; split; try Ens. }
      apply H15 in H16; apply Theorem55 in H16;
      apply Theorem49 in H13; auto.
      destruct H16; rewrite H16; auto.
   - split; try tauto; intros; destruct H12, H13.
     apply AxiomII in H13; destruct H13, H15.
     apply AxiomII in H15; destruct H15, H16.
     * apply AxiomII; split; Ens.
       assert ([u0, (En_f x y r s) [u0]] ∈ (En_f x y r s)).
       { apply Property_dom in H16; apply Property_Value; auto.
         apply H1 with v0; repeat split; auto. }
       exists (En_f x y r s) [u0]; apply AxiomII; split; Ens.
     * apply AxiomII in H16; destruct H16.
       assert ([u,v] ∈ 𝓤).
       { apply Theorem19; apply Theorem49; split; Ens. }
       apply H17 in H18; apply Theorem55 in H18;
       apply Theorem49 in H16; auto; destruct H18; subst v0.
       assert ([u0, (En_f x y r s) [u0]] ∈ (En_f x y r s)).
       { apply Property_Value; auto.
         generalize (classic (u0 ∈ dom( En_f x y r s))); intro.
         destruct H18; auto; absurd (Rrelation u0 r u); auto.
         apply H8; apply AxiomII; repeat split; Ens.
         apply AxiomII; split; Ens. }
       apply AxiomII; split; Ens.
       exists ((En_f x y r s)[u0]); apply AxiomII; split; Ens. }
assert (Section (ran(En_f x y r s ∪ [[u, v]])) s y).
{ unfold Section; split.
  - unfold Included; intros; apply AxiomII in H13; destruct H13, H14.
    apply AxiomII in H7; destruct H7 as [_ [H7 _]] .
```

```
    apply AxiomII in H14; destruct H14, H15.
    * apply Property_ran in H15;unfold Section in H2;apply H2;auto.
    * apply AxiomII in H15; destruct H15.
      assert ([u, v] ∈ 𝒰).
      { apply Theorem19; apply Theorem49; split; try Ens. }
      apply H16 in H17; apply Theorem55 in H17;
      apply Theorem49 in H14; auto; destruct H17; rewrite H18; auto.
  - split; try tauto; intros; destruct H13, H14.
    apply AxiomII in H14; destruct H14, H16.
    unfold Order_PXY in H3; destruct H3 as [_ [_ [H3 _]]].
    apply Theorem96 in H3; destruct H3 as [[_ H3] _].
    apply AxiomII in H16; destruct H16, H17.
    * apply AxiomII; split; Ens.
      assert ([((En_f x y r s) ⁻¹ ) [u0], u0] ∈ (En_f x y r s)).
      { assert (u0 ∈ ran( En_f x y r s)).
        { apply Property_ran in H17; apply H2 with v0; repeat split;
          auto. }
        pattern u0 at 2; rewrite Lemma96''' with (f:=(En_f x y r s));
        auto.
        apply Property_Value'; auto; rewrite <- Lemma96'''; auto. }
      exists ((En_f x y r s) ⁻¹ ) [u0];
      apply AxiomII; split; Ens.
    * apply AxiomII in H17; destruct H17.
      assert ([u,v] ∈ 𝒰).
      { apply Theorem19; apply Theorem49; split; Ens. }
      apply H18 in H19; apply Theorem55 in H19;
      apply Theorem49 in H16; auto; destruct H19; subst v0.
      assert ([((En_f x y r s) ⁻¹ ) [u0], u0] ∈ (En_f x y r s)).
      { generalize (classic (u0 ∈ ran( En_f x y r s))); intro;
        destruct H20.
        - pattern u0 at 2; rewrite Lemma96''' with(f:=(En_f x y r s);
          auto.
          apply Property_Value'; auto; rewrite <- Lemma96'''; auto.
        - absurd (Rrelation u0 s v); auto.
          apply H9; apply AxiomII; repeat split; Ens.
          apply AxiomII; split; Ens. }
      apply AxiomII; split; Ens.
      exists ((En_f x y r s) ⁻¹ ) [u0]; apply AxiomII; split; Ens. }
split.
- unfold Order_PXY; split; try tauto.
  split; try tauto; split; [idtac | tauto].
  unfold Order_Pr; intros; split; auto.
  split; try eapply Lemma99; eauto; try apply H.
  split; try eapply Lemma99; eauto; try apply H; intros.
  destruct H14, H15.
```

```
apply AxiomII in H14; destruct H14, H17.
apply AxiomII in H17; destruct H17 as [_ H17].
apply AxiomII in H15; destruct H15, H18.
apply AxiomII in H18; destruct H18 as [_ H18].
assert ([u,v] ∈ 𝓤) as H20.
{ apply Theorem19; apply Theorem49; split; Ens. }
destruct H17, H18.
* apply Property_dom in H17; apply Property_dom in H18;
  repeat rewrite Lemma99'; auto; Ens.
  unfold Order_PXY in H3; destruct H3 as [_ [_ [H3 _]]].
  unfold Order_Pr in H3; eapply H3; eauto.
* apply Property_dom in H17; rewrite Lemma99'; auto; Ens.
  apply AxiomII in H18; destruct H18.
  apply H19 in H20; apply Theorem55 in H20; destruct H20;
  apply Theorem49 in H18; auto.
  rewrite Lemma99''; auto; Ens.
  apply Lemma99''' with (y:=(ran( En_f x y r s))) (x:=y); auto.
  + apply Property_Value in H17; auto.
    double H17; apply Property_ran in H17.
    apply AxiomII;split;Ens;exists u0;apply AxiomII;split;Ens.
  + apply AxiomII in H7; destruct H7, H22;
    apply AxiomII in H23; tauto.
  + apply AxiomII in H7; tauto.
* apply Property_dom in H18.
  pattern ((En_f x y r s ∪ [[u, v]]) [v0]); rewrite Lemma99'; Ens.
  assert (u0 ∈ dom( En_f x y r s)).
  { unfold Section in H1; apply H1 with v0; split; auto.
    apply AxiomII in H17; destruct H17; apply H19 in H20.
    apply Theorem55 in H20; apply Theorem49 in H17; auto.
    destruct H20; rewrite H20; auto. }
  rewrite Lemma99'; Ens.
  unfold Order_PXY in H3; destruct H3 as [_ [_ [H3 _]]].
  unfold Order_Pr in H3; eapply H3; eauto.
* double H20.
  apply AxiomII in H17; destruct H17; apply H21 in H19.
  apply AxiomII in H18; destruct H18; apply H22 in H20.
  apply Theorem55 in H20; destruct H20;
  apply Theorem49 in H18; auto.
  apply Theorem55 in H19; destruct H19;
  apply Theorem49 in H17; auto.
  subst u0 v0.
  destruct H as [H _]; apply Theorem88 in H; destruct H as [_ H].
  apply Property_Asy with (u:=u) in H; auto; contradiction.
- assert (Ensemble ([u,v])). { apply Theorem49; split; Ens. }
  split.
```

```
    * apply AxiomII; split; Ens; exists v; apply AxiomII; split; Ens.
      right; apply AxiomII; split; auto.
    * apply AxiomII; split; Ens; right; apply AxiomII; split; auto.
Qed.

Hint Resolve Theorem99 : set.
```

在某种情况下, 可以确定定理 99 结论中的两种结果会出现哪一种, 因为如果 x 是集而 y 不是集, 则根据代换公理 V, f 的值域 $=y$ 是不可能的.

定理 100　*r 良序 x, s 良序 y, x 是集, y 不是集 $\Longrightarrow \exists! f$, f 是函数, f 在 x 和 y 中保序, f 的定义域 $=x$.*

```
Theorem Theorem100 : ∀ r s x y,
  WellOrdered r x /\ WellOrdered s y -> Ensemble x -> ~ Ensemble y ->
  ∃ f, Function f /\ Order_PXY f x y r s /\ dom( f) = x.
Proof.
  intros; destruct H.
  generalize (Lemma_y _ _ H H2); intro.
  apply Theorem99 in H3; destruct H3, H3, H4.
  exists x0; split; auto; split; auto; destruct H5; auto.
  unfold Order_PXY in H4; destruct H4 as [_ [_ [_ [H4 _]]]].
  unfold Section in H4; destruct H4; apply Theorem33 in H4; auto.
  apply AxiomV in H4; auto; rewrite H5 in H4; contradiction.
Qed.

Theorem Theorem100' : ∀ r s x y,
  WellOrdered r x /\ WellOrdered s y -> Ensemble x -> ~ Ensemble y ->
  ∀ f, Function f /\ Order_PXY f x y r s /\ dom(f) = x ->
  ∀ g, Function g /\ Order_PXY g x y r s /\ dom(g) = x -> f = g.
Proof.
  intros; destruct H, H2, H5, H3, H7; unfold Order_PXY in H5, H7.
  destruct H5 as [_ [_ H5]], H5, H9, H7 as [_ [_ H7]], H7, H11.
  generalize (Lemma_y _ _ H5 H7); intro.
  apply (Theorem97 f g r s x y) in H13; auto; destruct H13.
  - apply Theorem27; split; auto.
    unfold Included; intros.
    rewrite Theorem70; rewrite Theorem70 in H14; auto.
    PP H14 a b; double H15; rewrite <- Theorem70 in H15; auto.
    apply AxiomII_P in H16; destruct H16.
    apply AxiomII_P; split; auto; rewrite H17 in *.
    assert ([a,f[a]] ∈ f).
    { apply Property_Value; auto; subst x.
      apply Property_dom in H15; rewrite <- H8; auto. }
```

```
    apply H13 in H18; eapply H3; eauto.
  - apply Theorem27; split; auto.
    unfold Included; intros.
    rewrite Theorem70; rewrite Theorem70 in H14; auto.
    PP H14 a b; double H15; rewrite <- Theorem70 in H15; auto.
    apply AxiomII_P in H16; destruct H16.
    apply AxiomII_P; split; auto; rewrite H17 in *.
    assert ([a,g[a]] ∈ g).
    { apply Property_Value; auto; subst x.
      apply Property_dom in H15; rewrite H8; auto. }
    apply H13 in H18; eapply H2; eauto.
Qed.

Hint Resolve Theorem100 Theorem100' : set.

End A7.

Export A7.
```

3.8 序　　数

正则性公理 VII (VII Axiom of regularity)　$x \neq 0 \implies (\exists y, y \in x, x \cap y = 0)$.

```
Require Export A_7.

(* A.8 序数 *)

Module A8.

Axiom AxiomVII : ∀ x, x ≠ ∅ -> ∃ y, y ∈ x /\ x ∩ y = ∅.

Hint Resolve AxiomVII : set.
```

定理 101　$x \notin x$.

```
Theorem Theorem101 : ∀ x, x ∉ x.

Proof.
  intros; intro.
  assert ([x] ≠ ∅).
  { apply Property_NotEmpty; exists x; apply AxiomII; split; Ens.}
  apply AxiomVII in H0; destruct H0, H0.
  assert (x0 = x).
```

```
  { apply AxiomII in H0; destruct H0; apply H2; apply Theorem19; Ens.}
  subst x0.
  assert (x ∈ ([x] ∩ x)). { apply AxiomII; repeat split; Ens.}
  rewrite H1 in H2; generalize (Theorem16 x); intro; contradiction.
Qed.

Hint Resolve Theorem101 : set.
```

定理 102 $\sim\ (x \in y \wedge y \in x)$.

```
Theorem Theorem102 : ∀ x y,
   ~ (x ∈ y /\ y ∈ x).
Proof.
  intros; intro; destruct H.
  assert (\{ λ z, z = x \/ z =y \} ≠ ∅).
  { apply Property_NotEmpty; exists x; apply AxiomII; split; Ens. }
  apply AxiomVII in H1; destruct H1, H1;
  apply AxiomII in H1; destruct H1.
  destruct H3; subst x0.
  + assert (y ∈ (\{ λ z, z = x \/ z = y \} ∩ x)).
    { apply AxiomII; repeat split; Ens; apply AxiomII; split; Ens. }
    rewrite H2 in H3; generalize (Theorem16 y); intro; contradiction.
  + assert (x ∈ (\{ λ z, z = x \/ z = y \} ∩ y)).
    { apply AxiomII; repeat split; Ens; apply AxiomII; split; Ens. }
    rewrite H2 in H3; generalize (Theorem16 x); intro; contradiction.
Qed.

Hint Resolve Theorem102 : set.
```

定义 103 $E = \{(x, y) : x \in y\}$.

```
Definition E : Class := \{\ λ x y, x ∈ y \}\.

Hint Unfold E : set.
```

类 E 是 "E- **关系**". 若 $x \in y$ 同时 y 不是一个集, 则由定理 54 知, $(x, y) = \mathcal{U}$ 且 $(x, y) \notin E$.

定理 104 E 不是集.

```
Lemma Lemma104 : ∀ a b c,
  a ∈ b -> b ∈ c -> c ∈ a -> False.
Proof.
  intros.
  assert (\{ λ x, x = a \/ x =b \/ x = c \} ≠ ∅).
  { apply Property_NotEmpty; exists a; apply AxiomII; split; Ens. }
```

```
  apply AxiomVII in H2; destruct H2, H2;
  apply AxiomII in H2; destruct H2.
  destruct H4 as [H4 | [H4 | H4]]; subst x.
  + assert (c ∈ (\{ λ x, x = a \/ x =b \/ x = c \} ∩ a)).
    { apply AxiomII; repeat split; Ens; apply AxiomII; split; Ens. }
    rewrite H3 in H4; generalize (Theorem16 c); intro; contradiction.
  + assert (a ∈ (\{ λ x, x = a \/ x =b \/ x = c \} ∩ b)).
    { apply AxiomII; repeat split; Ens; apply AxiomII; split; Ens. }
    rewrite H3 in H4; generalize (Theorem16 a); intro; contradiction.
  + assert (b ∈ (\{ λ x, x = a \/ x =b \/ x = c \} ∩ c)).
    { apply AxiomII; repeat split; Ens; apply AxiomII; split; Ens. }
    rewrite H3 in H4; generalize (Theorem16 b); intro; contradiction.
Qed.

Theorem Theorem104 : ~ Ensemble E.
Proof.
  intro; generalize (Theorem42 _ H); intro.
  assert (E ∈ [E]).
  { apply AxiomII; split; auto. }
  assert ([E, [E]] ∈ E).
  { apply AxiomII_P; split; auto.
     apply Theorem49; tauto. }
  assert ([E] ∈ [E, [E]]).
  { apply AxiomII; split; Ens; left; apply AxiomII; split; auto. }
  eapply Lemma104; eauto.
Qed.

Hint Resolve Theorem104 : set.
```

定义 105 x 是充满的 $\Longleftrightarrow (y \in x \Longrightarrow y \subset x)$.

```
Definition full x : Prop := ∀ m, m ∈ x -> m ⊂ x.

Corollary Property_Full : ∀ x,
  full x <-> (∀ u v, v ∈ x /\ u ∈ v -> u ∈ x).
Proof.
  intros; split; intros.
  - unfold full in H; destruct H0; apply H in H0; auto.
  - unfold full;intros;unfold Included;intros;apply H with m;tauto.
Qed.

Hint Unfold full : set. Hint Resolve Property_Full : set.
```

x 是 “**充满的**” 当且仅当每个 x 元的元是 x 的元.

定义 106 x 是序 $\Longleftrightarrow E$ 连接 $x \wedge x$ 是充满的.

```
Definition Ordinal x : Prop := Connect E x /\ full x.

Hint Unfold Ordinal : set.
```

x 是 "**序**", 当且仅当已给 x 的两个元一个是另一个的元, 并且每一个 x 的元的元属于 x.

定理 107　x *是序* $\Longrightarrow E$ *良序* x.

```
Theorem Theorem107 : ∀ x,
  Ordinal x -> WellOrdered E x.
Proof.
  intros; unfold Ordinal in H; destruct H.
  unfold WellOrdered; intros; split; auto; intros; destruct H1.
  apply AxiomVII in H2; destruct H2, H2.
  exists x0; unfold FirstMember; intros.
  split; auto; intros; intro.
  unfold Rrelation in H5; apply AxiomII_P in H5; destruct H5.
  assert (y0 ∈ (y ∩ x0)). { apply AxiomII; split; Ens. }
  rewrite H3 in H7; generalize (Theorem16 y0); intro; contradiction.
Qed.

Hint Resolve Theorem107 : set.
```

定理 108　x *是序*, $y \subset x, y \neq x, y$ *是充满的* $\Longrightarrow y \in x$.

```
Theorem Theorem108 : ∀ x y,
  Ordinal x -> y ⊂ x -> y ≠ x -> full y -> y ∈ x.
Proof.
  intros.
  assert (Section y E x).
  { apply Theorem107 in H; unfold Section; intros.
    split; auto; split; auto; intros; destruct H3, H4.
    unfold Rrelation in H5; apply AxiomII_P in H5; destruct H5.
    unfold full in H2; apply H2 in H4; auto. }
  generalize (Lemma_y _ _ H3 H1); intro.
  apply Theorem91 in H4; destruct H4, H4.
  assert (x0 = \{ λ u, u ∈ x /\ Rrelation u E x0 \}).
  { apply AxiomI; split; intros; AssE z.
    + apply AxiomII; split; auto.
      unfold Ordinal in H; destruct H.
      double H4; unfold full in H8; apply H8 in H4.
      split; auto; apply AxiomII_P; split; auto.
      apply Theorem49; split; Ens.
    + apply AxiomII in H6; destruct H6, H8.
      unfold Rrelation in H9; apply AxiomII_P in H9; tauto. }
```

```
  rewrite <- H6 in H5; subst x0; auto.
Qed.

Hint Resolve Theorem108 : set.
```

定理 109 x 是序 $\wedge$ y 是序 $\implies$ $y \subset x \vee x \subset y$.

```
Lemma Lemma109 : ∀ x y,
  Ordinal x /\ Ordinal y -> full (x ∩ y).
Proof.
  intros; destruct H; unfold Ordinal in H, H0; destruct H, H0.
  unfold full in *; intros; apply AxiomII in H3; destruct H3, H4.
  apply H1 in H4; apply H2 in H5.
  unfold Included; intros; apply AxiomII; repeat split; Ens.
Qed.

Lemma Lemma109' : ∀ x y,
  Ordinal x /\ Ordinal y -> ((x ∩ y) = x) \/ ((x ∩ y) ∈ x).
Proof.
  intros; generalize (classic ((x ∩ y) = x)); intro;
  destruct H0; try tauto.
  assert ((x ∩ y) ⊂ x).
  { unfold Included; intros; apply Theorem4' in H1; tauto. }
  elim H; intros; apply Lemma109 in H.
  eapply Theorem108 in H2; eauto.
Qed.

Theorem Theorem109 : ∀ x y,
  Ordinal x /\ Ordinal y -> x ⊂ y \/ y ⊂ x.
Proof.
  intros; elim H; intros; generalize (Lemma_y _ _ H1 H0); intro.
  apply Lemma109' in H; apply Lemma109' in H2; destruct H.
  - apply Theorem30 in H; tauto.
  - destruct H2.
    + apply Theorem30 in H2; tauto.
    + assert ((x ∩ y) ∈ (x ∩ y)).
      { rewrite Theorem6' in H2; apply AxiomII; repeat split; Ens. }
      apply Theorem101 in H3; elim H3.
Qed.

Hint Resolve Theorem109 : set.
```

定理 110 x 是序 $\wedge$ y 是序 $\implies y \in x \vee x \in y \vee x = y$.

```
Theorem Theorem110 : ∀ x y,
```

```
  Ordinal x /\ Ordinal y -> x ∈ y \/ y ∈ x \/ x = y.
Proof.
  intros; generalize (classic (x = y)); intro; destruct H0; try tauto.
  elim H; intros; apply Theorem109 in H; destruct H.
  - left; unfold Ordinal in H1; destruct H1; eapply Theorem108; eauto.
  - right; left; unfold Ordinal in H2; destruct H2.
    eapply Theorem108; eauto; intro; auto.
Qed.

Hint Resolve Theorem110 : set.
```

定理 111　x 是序 $\wedge\, y \in x \implies y$ 是序.

```
Theorem Theorem111 : ∀ x y,
  Ordinal x /\ y ∈ x -> Ordinal y.
Proof.
  intros; destruct H; double H; unfold Ordinal in H; destruct H.
  assert (Connect E y).
  { unfold Connect; intros; unfold Ordinal in H1; apply H1 in H0.
    unfold Connect in H; destruct H3; apply H; auto. }
  unfold Ordinal; split; auto.
  unfold full; intros; unfold Included; intros.
  apply Theorem107 in H1; unfold Ordinal in H1.
  assert (y ⊂ x); auto; assert (m ∈ x); auto.
  assert (m ⊂ x); auto; assert (z ∈ x); auto.
  apply Theorem88 in H1; destruct H1.
  unfold Transitive in H1; specialize H1 with z m y.
  assert (Rrelation z E y).
  { apply H1; repeat split; Ens.
    + unfold Rrelation; apply AxiomII_P; split; auto.
      apply Theorem49; split; Ens.
    + unfold Rrelation; apply AxiomII_P; split; auto.
      apply Theorem49; split; Ens. }
  unfold Rrelation in H11; apply AxiomII_P in H11; tauto.
Qed.

Hint Resolve Theorem111 : set.
```

定义 112　$R = \{x : x$ 是序$\}$.

```
Definition R : Class := \{ λ x, Ordinal x \}.

Hint Unfold R : set.
```

定理 113[①]　R 是序但不是集.

```
Lemma Lemma113 : ∀ u v,
  Ensemble u -> Ensemble v -> Ordinal u /\ Ordinal v ->
  (Rrelation u E v \/ Rrelation v E u \/ u = v) .
Proof.
  intros; apply Theorem110 in H1; repeat split.
  destruct H1 as [H1 | [H1 | H1]].
  * left; unfold Rrelation; apply AxiomII_P; split; Ens.
    apply Theorem49; auto.
  * right; left; apply AxiomII_P; split; Ens.
    apply Theorem49; auto.
  * right; right; auto.
Qed.

Theorem Theorem113 : Ordinal R /\ ~ Ensemble R.
Proof.
  intros.
  assert (Ordinal R).
  { unfold Ordinal; intros; split.
    - unfold Connect; intros; destruct H.
      apply AxiomII in H; destruct H; apply AxiomII in H0; destruct H0.
      generalize (Lemma_y _ _ H1 H2); intro; apply Lemma113; auto.
    - unfold full; intros; apply AxiomII in H; destruct H.
      unfold Included; intros; apply AxiomII; split; Ens.
      eapply Theorem111; eauto. }
  split; auto; intro.
  assert (R ∈ R).
  { apply AxiomII; split; auto. }
  apply Theorem101 in H1; auto.
Qed.

Hint Resolve Theorem113 : set.
```

由定理 110, R 是仅有的非集的序.

定理 114　R 的每一个 E-截片是序.

```
Theorem Theorem114 : ∀ x,
  Section x E R -> Ordinal x.
Proof.
  intros.
  generalize (classic (x = R)); intro; destruct H0.
  - rewrite H0; apply Theorem113.
```

① 这个定理实质上是 Burali-Forti 悖论的叙述——在历史上是直观集论的第一个悖论 [41].

```
  - generalize (Lemma_y _ _ H H0); intro.
    apply Theorem91 in H1; destruct H1, H1.
    assert (x0 = \{ λ u, u ∈ R /\ Rrelation u E x0 \}).
    { apply AxiomI; split; intros.
      + apply AxiomII; repeat split; Ens.
        * apply AxiomII in H1; destruct H1.
          apply AxiomII; split; Ens; eapply Theorem111; eauto.
        * unfold Rrelation; apply AxiomII_P; split; auto.
          apply Theorem49; Ens.
      + apply AxiomII in H3; destruct H3, H4.
        unfold Rrelation in H5; apply AxiomII_P in H5; tauto. }
    subst x; rewrite H3 in H1; apply AxiomII in H1; tauto.
Qed.

Corollary Property114 : ∀ x,
  Ordinal x -> Section x E R.
Proof.
  intros; unfold Section; split.
  - unfold Included; intros; apply AxiomII; split; try Ens.
    eapply Theorem111; eauto.
  - split; intros; try apply Theorem107; try apply Theorem113.
    destruct H0, H1; unfold Ordinal in H2; apply AxiomII_P in H2;
    destruct H2.
    unfold Ordinal in H; destruct H; apply H4 in H1; auto.
Qed.

Hint Resolve Theorem114 Property114 : set.
```

定义 115　x 是序数 $\Longleftrightarrow x \in R$.

```
Definition Ordinal_Number x : Prop := x ∈ R.

Hint Unfold Ordinal_Number : set.
```

注意, 这里 “**序数**” 的概念不要与前面定义 106 中的 “序” 混淆. 因为序数属于 R, 所以序数一定是序且是集.

定义 116　$x < y \Longleftrightarrow x \in y$.

```
Definition Less x y : Prop := x ∈ y.

Notation "x ≺ y" := (Less x y)(at level 67, left
  associativity).

Hint Unfold Less : set.
```

定义 117 $x \leqslant y \iff x \in y \vee x = y$.

```
Definition LessEqual x y : Prop := x ∈ y \/ x = y.

Notation "x ⪯ y" := (LessEqual x y)(at level 67, left
  associativity).

Hint Unfold LessEqual : set.
```

定理 118 x 是序 $\wedge\ y$ 是序 $\implies (x \subset y \iff x \leqslant y)$.

```
Theorem Theorem118 : ∀ x y,
  Ordinal x /\ Ordinal y -> (x ⊂ y <-> x ⪯ y).
Proof.
  intros; destruct H; split; intros.
  - unfold LessEqual.
    generalize (classic (x = y)); intro; destruct H2; try tauto.
    unfold Ordinal in H; destruct H.
    left; apply Theorem108; auto.
  - unfold LessEqual in H1; destruct H1.
    + unfold Ordinal in H0; destruct H0; auto.
    + rewrite H1; auto; unfold Included; intros; auto.
Qed.

Hint Resolve Theorem118 : set.
```

定理 119 x 是序 $\implies x = \{y : y \in R \wedge y < x\}$.

```
Theorem Theorem119 : ∀ x,
  Ordinal x -> x = \{ λ y, y ∈ R /\ y ≺ x \}.
Proof.
  intros; apply AxiomI; split; intros.
  - apply AxiomII; repeat split; Ens.
    apply AxiomII; split; Ens; eapply Theorem111; eauto.
  - apply AxiomII in H0; destruct H0, H1; auto.
Qed.

Hint Resolve Theorem119 : set.
```

定理 120 $x \subset R \implies (\bigcup x)$ 是序.

```
Theorem Theorem120 : ∀ x,
  x ⊂ R -> Ordinal (⋃ x).
Proof.
  intros; red; split.
```

```
  - unfold Connect; intros; destruct H0.
    apply AxiomII in H0; apply AxiomII in H1;
    destruct H0, H2, H2, H1, H4, H4.
    apply H in H3; apply H in H5.
    apply AxiomII in H3; destruct H3; apply AxiomII in H5; destruct H5.
    assert (Ordinal u). { eapply Theorem111; eauto. }
    assert (Ordinal v). { eapply Theorem111; eauto. }
    generalize (Lemma_y _ _ H8 H9); intro; apply Lemma113; auto.
  - apply Property_Full; intros; destruct H0.
    apply AxiomII in H0; destruct H0, H2, H2.
    apply AxiomII; split; Ens.
    exists x0; split; auto.
    apply H in H3; apply AxiomII in H3; destruct H3 as [_ H3].
    unfold Ordinal in H3; destruct H3; apply H4 in H2; auto.
Qed.

Hint Resolve Theorem120 : set.
```

可以看到, 如果 x 是 R 的子集, 则序 $\bigcup x$ 是大于等于 x 的每个元的第一个序, 同时 $\bigcup x$ 是一个集当且仅当 x 是一个集.

定理 121　$x \subset R \wedge x \neq 0 \implies (\bigcap x) \in x.$

```
Lemma Lemma121 : ∀ x,
  x ⊂ R /\ x ≠ ∅ -> FirstMember (⋂ x) E x.
Proof.
  intros; destruct H; generalize (Theorem113); intro; destruct H1.
  apply Theorem107 in H1; unfold WellOrdered in H1; destruct H1.
  generalize (Lemma_y _ _ H H0); intro; apply H3 in H4; destruct H4.
  double H4; unfold FirstMember in H4; destruct H4.
  assert ((⋂ x) = x0).
  { apply AxiomI; split; intros.
    + apply AxiomII in H7; destruct H7; apply H8; auto.
    + apply AxiomII; split; Ens; intros.
      assert (~ Rrelation y E x0); auto.
      assert (Ordinal x0). {apply H in H4; apply AxiomII in H4; tauto.}
      assert (Ordinal y). { apply H in H8; apply AxiomII in H8; tauto.}
      generalize (Lemma_y _ _ H10 H11); intro; apply Theorem110 in H12.
      destruct H12 as [H12 | [H12 | H12]].
      * apply H in H8; apply AxiomII in H8; destruct H8 as [_ H8].
        unfold Ordinal in H8; destruct H8;
        generalize (Property_Full y); intro.
        destruct H14; eapply H14; eauto.
      * elim H9; unfold Rrelation; apply AxiomII_P; split; auto.
        apply Theorem49; Ens.
      * subst x0; auto. }
```

```
  rewrite H7 ; auto.
Qed.

Theorem Theorem121 : ∀ x,
  x ⊂ R /\ x ≠ ∅ -> (⋂ x) ∈ x.
Proof.
  intros; apply Lemma121 in H.
  unfold FirstMember in H; tauto.
Qed.

Hint Resolve Theorem121 : set.
```

诚然, 若 $x \subset R$, 则 $\bigcap x$ 是 x 的 E-首元.

定义 122 $x+1=x\cup\{x\}$.

```
Definition PlusOne x := x ∪ [x].

Hint Unfold PlusOne : set.
```

定理 123 $x\in R \implies x+1$ 是 $\{y:y\in R\wedge x<y\}$ 的 E- 首元.

```
Lemma Lemma123 : ∀ x, x ∈ R -> (PlusOne x) ∈ R.
Proof.
  intros; apply AxiomII; split.
  - apply AxiomIV; split; Ens; apply Theorem42; Ens.
  - unfold Connect; split.
    * unfold Connect; intros; destruct H0.
      apply AxiomII in H0; apply AxiomII in H1;destruct H0, H1, H2, H3.
      -- apply AxiomII in H; destruct H as [_ H].
          assert (Ordinal u). { eapply Theorem111; eauto. }
          assert (Ordinal v). { eapply Theorem111; eauto. }
          generalize (Lemma_y _ _ H4 H5); intro; apply Lemma113; auto.
      -- apply AxiomII in H3; destruct H3.
          AssE x; apply Theorem19 in H5; apply H4 in H5; subst v.
          left; unfold Rrelation; apply AxiomII_P; split; auto.
          apply Theorem49; tauto.
      -- apply AxiomII in H2; destruct H2.
          AssE x; apply Theorem19 in H5; apply H4 in H5; subst u.
          right; left; unfold Included; apply AxiomII_P; split; auto.
          apply Theorem49; tauto.
      -- AssE x; apply Theorem19 in H4; double H4.
          apply AxiomII in H2; destruct H2; apply H6 in H4.
          apply AxiomII in H3; destruct H3; apply H7 in H5.
          subst u; subst v; tauto.
    * unfold full; intros; unfold Included; intros.
```

```
      apply AxiomII in H; apply AxiomII in H0; destruct H, H0.
      apply AxiomII; split; Ens; destruct H3.
      -- unfold Ordinal in H2; destruct H2.
         unfold full in H4; left; eapply H4; eauto.
      -- apply AxiomII in H3; destruct H3.
         apply Theorem19 in H; apply H4 in H; subst m; tauto.
Qed.

Theorem Theorem123 : ∀ x,
  x ∈ R -> FirstMember (PlusOne x) E (\{ λ y, (y ∈ R /\ x ≺ y) \}).
Proof.
  intros; unfold FirstMember; split; intros.
  - apply AxiomII; repeat split.
    + unfold Ensemble; exists R; apply Lemma123; auto.
    + apply Lemma123; auto.
    + unfold Less; intros; apply AxiomII; split; Ens.
      right; apply AxiomII; split; Ens.
  - intro; apply AxiomII in H0; destruct H0, H2.
    unfold Rrelation in H1; apply AxiomII_P in H1; destruct H1.
    apply AxiomII in H4; destruct H4; unfold Less in H3; destruct H5.
    + eapply Theorem102; eauto.
    + AssE x; apply Theorem19 in H6; apply AxiomII in H5; destruct H5.
      apply H7 in H6; subst y; eapply Theorem101; eauto.
Qed.

Hint Resolve Theorem123 : set.
```

定理 124　$x \in R \implies \bigcup(x+1) = x.$

```
Theorem Theorem124 : ∀ x,
  x ∈ R -> ⋃ (PlusOne x) = x.
Proof.
  intros; apply AxiomI; split; intros.
  - apply AxiomII in H0; destruct H0, H1, H1.
    apply AxiomII in H2; destruct H2, H3.
    + apply AxiomII in H; destruct H, H4.
      generalize (Property_Full x); intro; destruct H6.
      apply H6 with (u := z) (v := x0) in H5; tauto.
    + apply AxiomII in H3; destruct H3.
      rewrite <- H4; auto; try (apply Theorem19; Ens).
  - apply AxiomII; split; Ens; exists x; split; auto.
    apply AxiomII; split; Ens; right; apply AxiomII; Ens.
Qed.

Hint Resolve Theorem124 : set.
```

定义 125 $f|x = f \cap (x \times \mathcal{U})$.

```
Definition Restriction f x : Class := f ∩ (x × 𝒰).

Notation "f | ( x )" := (Restriction f x)(at level 30).

Hint Unfold Restriction : set.
```

这个定义仅只在 f 是一个关系时才使用, 在这种情况下, $f|x$ 是一个关系同时称为 f 在 x 上的 "**限制**".

定理 126 f 是函数 $\Longrightarrow$ $f|x$ 是函数, $(f|x)$ 的定义域 $= x \cap (f$ 的定义域 $)$, $(\forall y \in (f|x)$ 的定义域, $(f|x)(y) = f(y))$.

```
Theorem Theorem126 : ∀ f x,
  Function f -> Function (f | (x)) /\ dom( f | (x)) = x ∩ dom( f) /\
  (∀ y, y ∈ dom( f | (x)) -> (f | (x)) [y] = f [y]).
Proof.
  intros; repeat split; intros.
  - unfold Relation; intros; apply AxiomII in H0; destruct H0, H1.
    PP H2 a b; eauto.
  - destruct H0; apply AxiomII in H0; destruct H0 as [_ [H0 _]].
    apply AxiomII in H1; destruct H1 as [_ [H1 _]].
    unfold Function in H; eapply H; eauto.
  - apply AxiomI; split; intros.
    + apply AxiomII in H0; destruct H0, H1.
      apply AxiomII in H1; destruct H1, H2.
      apply Property_dom in H2; apply AxiomII_P in H3.
      apply AxiomII; split; tauto.
    + apply AxiomII in H0; destruct H0, H1; apply AxiomII; split; auto.
      apply Property_Value in H2; auto.
      exists f[z]; apply AxiomII; repeat split; Ens.
      apply AxiomII_P; repeat split; Ens; apply Theorem19.
      apply Property_ran in H2; Ens.
  - apply AxiomI; split; intros.
    + apply AxiomII in H1; destruct H1.
      apply AxiomII in H0; destruct H0, H3.
      apply AxiomII in H3; destruct H3, H4.
      apply Property_dom in H4; apply H2.
      assert (Ensemble f[y]). {apply Theorem19; apply Theorem69; auto.}
      apply AxiomII; split; Ens; apply AxiomII; repeat split.
      * apply Theorem49; auto.
      * apply Property_Value in H4; auto.
      * apply AxiomII_P in H5; apply Theorem19 in H6.
        apply AxiomII_P; repeat split; try tauto;
```

```
      try apply Theorem49; Ens.
    + apply AxiomII in H1; destruct H1;
      apply AxiomII; split; auto; intros.
      apply AxiomII in H3; destruct H3; apply AxiomII in H4.
      apply H2; apply AxiomII; split; tauto.
Qed.

Hint Resolve Theorem126 : set.
```

定理 127 (f 是函数, f 的定义域是序, ($\forall u \in f$ 的定义域, $f(u) = g(f|u)$)) $\wedge$ (h 是函数, h 的定义域是序, ($\forall u \in h$ 的定义域, $h(u) = g(h|u)$)) $\Longrightarrow (h \subset f) \vee (f \subset h)$.

```
Lemma Lemma127 : ∀ f h,
  dom( f ) ⊂ dom( h ) -> Function f -> Function h ->
  \{ λ a,a∈(dom(f)∩dom(h))/\f[a]≠h[a]\}=∅ -> f ⊂ h.
Proof.
  intros.
  unfold Included; intros; rewrite Theorem70 in H3; auto; PP H3 a b.
  double H4; rewrite<-Theorem70 in H4; auto; apply Property_dom in H4.
  apply AxiomII_P in H5; destruct H5.
  rewrite Theorem70; auto; apply AxiomII_P; split; auto; rewrite H6.
  generalize (classic (f[a] = h[a])); intro; destruct H7; auto.
  assert (a ∈ \{ λ a,a∈(dom(f)∩dom(h))/\f[a]≠h[a]\}).
  { apply Theorem30 in H; rewrite H; apply AxiomII; split; Ens. }
  rewrite H2 in H8; generalize (Theorem16 a); contradiction.
Qed.

Theorem Theorem127 : ∀ f h g,
  Function f -> Ordinal dom( f ) ->
  (∀ u0, u0 ∈ dom( f ) -> f [u0] = g [f | (u0)]) ->
  Function h -> Ordinal dom( h ) ->
  (∀ u1, u1 ∈ dom( h ) -> h [u1] = g [h | (u1)]) ->
  h ⊂ f \/ f ⊂ h.
Proof.
  intros; generalize (Lemma_y_ _H0 H3); intro; apply Theorem109 in H5.
  generalize (classic (\{λ a,a∈(dom(f)∩dom(h))/\f[a]≠h[a]\}=∅));
  intro; destruct H6.
  - destruct H5.
    + right; apply Lemma127; auto.
    + left; rewrite Lemma97'' in H6; apply Lemma127; auto.
  - assert (exists u, FirstMember u E \{λ a,a∈(dom(f)∩dom(h))/\
    f[a]≠h [a]\}).
    { apply Theorem107 in H0; unfold WellOrdered in H0;
      apply H0; split; auto.
      unfold Included; intros; apply AxiomII in H7; destruct H7, H8.
```

```
    apply AxiomII in H8; tauto. }
  destruct H7 as [u H7]; unfold FirstMember in H7; destruct H7.
  apply AxiomII in H7; destruct H7, H9.
  apply AxiomII in H9; destruct H9 as [_[H9 H11]].
  generalize (H1 _ H9); generalize (H4 _ H11); intros.
  assert ((h | (u)) = (f | (u))).
  { apply AxiomI; intros; split; intros.
    + apply AxiomII in H14; destruct H14, H15.
      apply AxiomII; repeat split; auto; PP H16 a b.
      apply AxiomII_P in H17; destruct H17 ,H18.
      generalize H15 as H22; intro; apply Property_dom in H22.
      rewrite Theorem70 in H15; auto; rewrite Theorem70; auto.
      apply AxiomII_P in H15; destruct H15;
      apply AxiomII_P; split; auto.
      rewrite H20; symmetry.
      generalize (classic (f [a]=h [a])); intro; destruct H21; auto.
      assert (a ∈ \{ λ a, a ∈ (dom( f ) ∩ dom( h )) /\
      f [a] ≠ h [a] \}).
      { apply AxiomII; repeat split; Ens;
        apply AxiomII; repeat split; Ens.
        unfold Ordinal in H0; destruct H0.
        unfold full in H23; apply H23 in H9; auto. }
      apply H8 in H23; elim H23; unfold Rrelation, E.
      apply AxiomII_P; split; auto; apply Theorem49; split; Ens.
    + apply AxiomII in H14; destruct H14, H15.
      apply AxiomII; repeat split; auto; PP H16 a b.
      apply AxiomII_P in H17; destruct H17 ,H18.
      generalize H15 as H22; intro; apply Property_dom in H22.
      rewrite Theorem70 in H15; auto; rewrite Theorem70; auto.
      apply AxiomII_P in H15; destruct H15;
      apply AxiomII_P; split; auto.
      rewrite H20; symmetry.
      generalize (classic (f [a]=h [a])); intro; destruct H21; auto.
      assert (a ∈ \{ λ a, a ∈ (dom( f ) ∩ dom( h )) /\ f [a] ≠
      h [a] \}).
      { apply AxiomII; repeat split; Ens;
        apply AxiomII; repeat split; Ens.
        unfold Ordinal in H3; destruct H3.
        unfold full in H23; apply H23 in H11; auto. }
      apply H8 in H23; elim H23; unfold Rrelation, E.
      apply AxiomII_P; split; auto; apply Theorem49; split; Ens. }
 rewrite <- H14 in H13; rewrite <- H12 in H13; contradiction.
Qed.

Hint Resolve Theorem127 : set.
```

定理 128　对于每一个 g 存在唯一的函数 f 使得 f 的定义域是序, 并且对每一个序数 x 有 $f(x) = g(f|x)$.

```
Definition En_f' g := \{\ λ u v, u ∈ R /\ (exists h, Function h /\
  Ordinal dom( h ) /\ (∀ z, z ∈ dom( h ) -> h [z] = g [h | (z)] ) /\
  [u,v] ∈ h ) \}\.

Lemma Lemma128 : ∀ u v w,
  Ordinal u -> v ∈ u -> w ∈ v -> w ∈ u.
Proof.
  intros; unfold Ordinal in H; destruct H.
  unfold full in H2; eapply H2; eauto.
Qed.

Lemma Lemma128' : ∀ f x,
  Function f -> Ordinal dom( f ) -> Ordinal_Number x ->
  ~ x ∈ dom( f ) -> f | (x) = f.
Proof.
  intros; apply AxiomI; split; intros.
  - apply AxiomII in H3; tauto.
  - apply AxiomII; split; Ens; split; auto.
    unfold Function, Relation in H; destruct H as [H _].
    double H3; apply H in H4; destruct H4 as [a[b H4]]; rewrite H4 in*.
    clear H H4; apply AxiomII_P; split; Ens; split.
    + unfold Ordinal in H1; apply AxiomII in H1; destruct H1.
      assert (Ordinal dom(f) /\ Ordinal x); auto.
      apply Theorem110 in H4; apply Property_dom in H3; auto.
      destruct H4 as [H4 | [H4 | H4]]; try contradiction.
      * eapply Lemma128; eauto.
      * rewrite H4 in H3; auto.
    + apply Property_ran in H3; apply Theorem19; Ens.
Qed.

Theorem Theorem128 : ∀ g,
  ∃ f, Function f /\ Ordinal dom( f ) /\
  (∀ x, Ordinal_Number x -> f [x] = g [f | (x)]).
Proof.
  intros; exists (En_f' g).
  assert (Function (En_f' g)).
  { unfold Function; intros; split; intros.
    - unfold Relation; intros; PP H a b; eauto.
    - destruct H; apply AxiomII_P in H; apply AxiomII_P in H0.
      destruct H, H1, H2, H2, H3, H4, H0, H6, H7, H7, H8, H9.
      generalize (Theorem127 _ _ _ H2 H3 H4 H7 H8 H9); intro;
      destruct H11.
```

```
    + apply H11 in H10; eapply H2; eauto.
    + apply H11 in H5; eapply H7; eauto. }
split; auto.
- assert (Ordinal dom(En_f' g)).
  { apply Theorem114; unfold Section; intros; split.
    - unfold Included; intros; apply AxiomII in H0.
      destruct H0, H1; apply AxiomII_P in H1; tauto.
    - split; intros.
      + apply Theorem107; apply Theorem113.
      + destruct H0, H1; apply AxiomII in H1; destruct H1, H3.
        apply AxiomII_P in H3; destruct H3, H4, H5, H5, H6, H7.
        apply AxiomII_P in H2; destruct H2.
        apply Theorem49 in H2; destruct H2.
        apply AxiomII; split; auto; apply Property_dom in H8.
        assert (u ∈ dom( x0)). { eapply Lemma128; eauto. }
        exists (x0[u]); apply AxiomII_P.
        split; try apply Theorem49; split; auto.
        * apply Theorem19; apply Theorem69; auto.
        * exists x0; split; auto; split; auto; split; auto.
          apply Property_Value; auto. }
  split; intros; auto.
  generalize (classic (x ∈ dom(En_f' g))); intro; destruct H2.
  + apply AxiomII in H2; destruct H2, H3; apply AxiomII_P in H3.
    destruct H2, H3, H4, H5 as [h [H5 [H6 [H7 H8]]]].
    assert (h ⊂ En_f' g).
    { double H5; unfold Included; intros;
      unfold Function, Relation in H9.
      destruct H9 as [H9 _]; double H10; apply H9 in H11.
      destruct H11 as [a [b H11]]; rewrite H11 in *; clear H9 H11 z.
      apply AxiomII_P; split; try Ens; double H10;
      apply Property_dom in H9.
      split; try apply AxiomII; Ens; split; Ens.
      eapply Theorem111; eauto. }
    double H8; apply H9 in H10; double H8.
    apply Property_dom in H11; apply H7 in H11.
    double H8; apply Property_dom in H12; apply Property_dom in H8.
    apply Property_Value in H8; auto; apply Property_dom in H10.
    apply Property_Value in H10; auto; apply H9 in H8.
    assert (h [x] = (En_f' g) [x]). { eapply H; eauto. }
    rewrite <- H13; clear H13.
    assert (h | (x) = En_f' g | (x)).
    { apply AxiomI; split; intros;
      apply AxiomII in H13; destruct H13, H14.
     - apply AxiomII; repeat split; auto.
     - apply AxiomII; repeat split; auto; rewrite Theorem70; auto.
```

```
    PP H15 a b; apply AxiomII_P in H16;
    apply AxiomII_P; split; auto.
    destruct H16, H17; assert (a ∈ dom(h)).
    { eapply Lemma128; eauto. }
    apply Property_Value in H19; auto;
    apply H9 in H19; eapply H; eauto. }
  rewrite <- H13; auto.
+ generalize H2; intro; apply Theorem69 in H2; auto.
  rewrite (Lemma128' _ _ H H0 H1 H3).
  generalize (classic (En_f' g ∈ dom(g))); intro; destruct H4.
  * generalize Theorem113; intro; destruct H5 as [H5 _].
    apply Theorem107 in H5; unfold WellOrdered in H5; destruct H5.
    assert ((R ~ dom(En_f' g)) ⊂ R /\ (R ~ dom(En_f' g)) ≠ ∅).
    { split; try (red; intros; apply AxiomII in H7; tauto).
      intro; generalize (Property114 _ H0); intro.
      unfold Section in H8; destruct H8.
      apply Property_∅ in H8; apply H8 in H7.
      rewrite <- H7 in H3; contradiction. }
    apply H6 in H7; destruct H7 as [y H7].
    assert (((En_f' g) ∪ [[y,g[En_f' g]]]) ⊂ (En_f' g)).
    { unfold Included; intros; apply AxiomII in H8;
      destruct H8, H9; auto.
      assert (Ensemble ([y, g [En_f' g]])).
      { destruct H7; AssE y; apply Theorem69 in H4.
        apply Theorem19 in H4; apply Theorem49; tauto. }
      apply AxiomII in H9; destruct H9.
      rewrite H11; try apply Theorem19; auto.
      apply AxiomII_P; split; auto; split.
      - unfold FirstMember in H7; destruct H7;
        apply AxiomII in H7; tauto.
      - exists ((En_f' g) ∪ [[y,g[En_f' g]]]).
        assert (Function (En_f' g ∪ [[y, g [En_f' g]]])).
        { unfold Function; split; intros.
          - unfold Relation; intros; apply AxiomII in H12.
            destruct H12, H13; try PP H13 a b; eauto.
            apply AxiomII in H13; destruct H13;
            apply Theorem19 in H10.
            apply H14 in H10; eauto.
          - destruct H12; apply AxiomII in H12;
            destruct H12 as [_ H12].
            apply AxiomII in H13; destruct H13 as [_ H13].
            unfold FirstMember in H7; destruct H7.
            apply AxiomII in H7; destruct H7 as [_ [_ H7]].
            apply AxiomII in H7; destruct H7, H12, H13.
            + eapply H; eauto.
```

```
    + apply AxiomII in H13; destruct H13;
      apply Theorem19 in H10.
      apply H16 in H10; apply Theorem55 in H10;
      destruct H10; try apply Theorem49; auto;
      rewrite H10 in H12.
      apply Property_dom in H12; contradiction.
    + apply AxiomII in H12; destruct H12;
      apply Theorem19 in H10.
      apply H16 in H10;apply Theorem55 in H10;destruct H10;
      try apply Theorem49; auto; rewrite H10 in H13.
      apply Property_dom in H13; contradiction.
    + double H12;apply AxiomII in H12;apply AxiomII in H13.
      destruct H12, H13; double H10.
      apply Theorem19 in H10; apply H17 in H10.
      apply Theorem19 in H19; apply H18 in H19.
      apply Theorem55 in H10; destruct H10;
      apply Theorem49 in H12; auto.
      apply Theorem55 in H19; destruct H19;
      apply Theorem49 in H13; auto.
      rewrite H20, H21; auto. }
split; auto; split.
+ apply Theorem114; unfold Section; intros; split.
  * unfold Included; intros.
    apply AxiomII in H13; destruct H13, H14.
    apply AxiomII in H14; destruct H14, H15.
    -- apply Property_dom in H15; apply AxiomII.
       split; Ens; eapply Theorem111; eauto.
    -- apply AxiomII in H15; destruct H15;
       apply Theorem19 in H10.
       apply H16 in H10;apply Theorem55 in H10;destruct H10;
       try apply Theorem49; auto; destruct H7.
       apply AxiomII in H7; rewrite H10; tauto.
  * split; try (apply Theorem107;apply Theorem113);intros.
    destruct H13,H14;apply AxiomII in H14;destruct H14,H16.
    apply AxiomII in H16; destruct H16, H17.
    -- apply AxiomII; split; Ens.
       assert ([u, (En_f' g) [u]] ∈ (En_f' g)).
       {apply Property_Value;auto;apply Property_dom in H17.
        unfold Rrelation in H15; apply AxiomII_P in H15.
        destruct H15; eapply Lemma128; eauto. }
       exists ((En_f' g) [u]); apply AxiomII; split; Ens.
    -- assert ([u, (En_f' g) [u]] ∈ (En_f' g)).
       {apply Property_Value; auto.
        apply AxiomII in H17; destruct H17.
        apply Theorem19 in H10; apply H18 in H10.
```

```
        apply Theorem55 in H10;
        destruct H10; try apply Theorem49; auto.
        subst v; unfold FirstMember in H7; destruct H7.
        generalize (classic (u ∈ dom( En_f' g))); intro.
        destruct H20; auto.
        absurd (Rrelation u E y); auto; try apply H10.
        apply AxiomII; repeat split; Ens.
        apply AxiomII; split; Ens. }
       apply AxiomII; split; Ens; exists ((En_f' g) [u]).
       apply AxiomII; split; Ens.
+ split; intros.
  * apply Property_Value in H13; auto.
    apply AxiomII in H13; destruct H13, H14.
    -- apply AxiomII_P in H14; destruct H14, H15.
       destruct H16 as [h [H16 [H17 [H18 H19]]]].
       double H19; apply Property_dom in H20.
       rewrite Theorem70 in H19; auto.
       apply AxiomII_P in H19; destruct H19.
       assert (h ⊂ En_f' g).
       { unfold Included; intros; double H16.
         unfold Function, Relation in H23;
         destruct H23 as [H23 _].
         double H22; apply H23 in H24;
         destruct H24 as [a [b H24]].
         rewrite H24 in *; clear H23 H24; apply AxiomII_P.
         split;try Ens;double H22;apply Property_dom in H23.
         split; try apply AxiomII; Ens.
         split; try Ens; eapply Theorem111; eauto. }
       assert ((En_f' g ∪ [[y, g[En_f' g]]])|(z0) =
       En_f' g| (z0)).
       { unfold Restriction; rewrite Theorem6';
         rewrite Theorem8.
         assert ((z0) × 𝒰 ∩ [[y, g [En_f' g]]] = ∅).
         { apply AxiomI; split; intros.
           - apply AxiomII in H23; destruct H23, H24; auto.
             PP H24 a b; apply AxiomII_P in H26;
             destruct H26, H27.
             apply AxiomII in H25; destruct H25.
             apply Theorem19 in H10; apply H29 in H10.
             apply Theorem55 in H10;
             apply Theorem49 in H25; auto.
             destruct H10; rewrite H10 in H27.
             assert (y∈dom( h)). { eapply Lemma128; eauto.}
             apply Property_Value in H31; auto.
             apply H22 in H31; apply Property_dom in H31.
```

```
          unfold FirstMember in H7; destruct H7.
          apply AxiomII in H7; destruct H7, H33.
          apply AxiomII in H34;destruct H34;contradiction.
        - generalize (Theorem16 z1); contradiction. }
      rewrite H23, Theorem6, Theorem17;apply Theorem6'.}
    rewrite H21, H23.
    assert (h | (z0) = En_f' g | (z0)).
    { apply AxiomI; split; intros.
      - apply AxiomII in H24; destruct H24, H25.
        apply AxiomII; repeat split; auto.
      - apply AxiomII in H24; destruct H24, H25;
        apply AxiomII.
        repeat split; auto; rewrite Theorem70; auto.
        PP H26 a b;apply AxiomII_P in H27;apply AxiomII_P.
        split; auto; destruct H27 as [_ [H27 _]].
        assert (a∈dom(h)). {eapply Lemma128; eauto.}
        apply Property_Value in H28;auto;apply H22 in H28.
        eapply H; eauto. }
    rewrite <- H24; auto.
 -- apply AxiomII in H14; destruct H14.
    double H10; apply Theorem19 in H10; apply H15 in H10.
    apply Theorem55 in H10; apply Theorem49 in H13; auto.
    destruct H10; subst z0; rewrite H17.
    assert ((En_f' g ∪ [[y, g [En_f' g]]])|(y) =
    En_f' g|(y)).
    { apply AxiomI; split; intros.
      - apply AxiomII in H10; destruct H10, H18.
        apply AxiomII in H18; destruct H18, H20.
        + apply AxiomII; tauto.
        + PP H19 a b; apply AxiomII_P in H21;
          destruct H21, H22.
          apply AxiomII in H20; destruct H20.
          apply Theorem19 in H16; apply H24 in H16.
          apply Theorem55 in H16;
          apply Theorem49 in H21; auto.
          destruct H16; rewrite H16 in H22.
          generalize (Theorem101 y); intro; contradiction.
      - unfold Restriction; rewrite Theorem6', Theorem8.
        apply AxiomII; split; Ens; left;
        rewrite Theorem6'; Ens. }
    rewrite H10; unfold FirstMember in H7; destruct H7.
    apply AxiomII in H7; destruct H7, H19.
    apply AxiomII in H20; destruct H20;
    rewrite Lemma128'; auto.
* apply AxiomII; split; Ens; right; apply AxiomII; split;
```

```
            Ens. }
      unfold FirstMember in H7; destruct H7.
      assert (y ∈ dom(En_f' g ∪ [[y, g [En_f' g]]])).
      { apply AxiomII; split; Ens; exists g [En_f' g].
        assert (Ensemble ([y, g [En_f' g]])).
        { apply Theorem49; split; Ens.
          apply Theorem69 in H4; apply Theorem19; auto. }
        apply AxiomII; split; Ens; right; apply AxiomII; auto. }
      apply AxiomII in H7; destruct H7, H11; apply AxiomII in H12.
      destruct H12; elim H13; apply AxiomII in H10;destruct H10,H14.
      apply H8 in H14; apply Property_dom in H14; auto.
    * apply Theorem69 in H4; rewrite H2, H4; auto.
Qed.

Lemma Lemma128'' : ∀ f h,
  Function f -> Function h -> h ⊂ f -> f | (dom( h )) = h.
Proof.
  intros; apply AxiomI; split; intros.
  - apply AxiomII in H2; destruct H2, H3.
    PP H4 a b; apply AxiomII_P in H5; destruct H5, H6; double H3.
    rewrite Theorem70; rewrite Theorem70 in H3; auto.
    apply AxiomII_P in H3; destruct H3.
    apply AxiomII_P; split; Ens; rewrite H9 in *.
    apply Property_Value in H6; auto.
    apply H1 in H6; eapply H; eauto.
  - apply AxiomII; repeat split; Ens.
    rewrite Theorem70 in H2; auto.
    PP H2 a b; rewrite <- Theorem70 in H3; auto.
    apply AxiomII_P; repeat split; Ens.
    + apply Property_dom in H3; auto.
    + AssE [a,b]; apply Theorem49 in H4.
      apply Theorem19; tauto.
Qed.

Lemma Lemma128''' : ∀ h,
  Function h -> h | (dom( h )) = h.
Proof.
  intros; apply AxiomI; split; intros.
  - apply AxiomII in H0; tauto.
  - apply AxiomII; repeat split; Ens.
    rewrite Theorem70 in H0; auto.
    PP H0 a b; rewrite <- Theorem70 in H1; auto.
    apply AxiomII_P; repeat split; Ens.
    + apply Property_dom in H1; auto.
    + AssE [a,b]; apply Theorem49 in H2.
```

```
    apply Theorem19; tauto.
Qed.

Lemma Lemma128'''' : ∀ f g h,
  Function f -> Function h -> Ordinal dom( f ) ->
  Ordinal dom( h )-> (∀ x, Ordinal_Number x -> f [x]=g [ f | (x)]) ->
  (∀ x, Ordinal_Number x -> h [x] = g [ h | (x) ]) -> h ⊂ f -> h = f.
Proof.
  intros; generalize (Theorem110 _ _ (Lemma_y _ _ H1 H2)); intro.
  destruct H6 as [H8 | [H6 | H6]].
  + apply Property_Value in H8; auto.
    apply H5 in H8; apply Property_dom in H8.
    apply Theorem101 in H8; elim H8.
  + assert (Ordinal_Number dom( h )).
    { unfold Ordinal_Number; apply AxiomII; split; Ens. }
    double H7; apply H3 in H7; apply H4 in H8.
    rewrite Lemma128'' in H7; rewrite Lemma128''' in H8; auto.
    apply Theorem69 in H6; rewrite H7 in H6.
    generalize (Theorem101 dom( h )); intro.
    apply Theorem69 in H9; rewrite H8 in H9.
    rewrite H9 in H6; apply Theorem101 in H6; elim H6.
  + apply Theorem27; split; auto.
    unfold Included; intros.
    rewrite Theorem70; rewrite Theorem70 in H7; auto.
    PP H7 a b; double H8; rewrite <- Theorem70 in H8; auto.
    apply AxiomII_P in H9; destruct H9.
    apply AxiomII_P; split; auto; rewrite H10 in *.
    assert ([a,h[a]] ∈ h).
    { apply Property_Value; auto.
      apply Property_dom in H8; rewrite <- H6; auto. }
    apply H5 in H11; eapply H; eauto.
Qed.

Theorem Theorem128' : ∀ g,
  ∀ f, Function f /\ Ordinal dom(f) /\
  (∀ x, Ordinal_Number x -> f [x] = g [f | (x)]) ->
  ∀ h, Function h /\ Ordinal dom( h ) /\
  (∀ x, Ordinal_Number x -> h [x] = g [ h | (x) ]) -> f = h.
Proof.
  intros; destruct H, H0, H1, H2.
  assert (forall u : Class, u ∈ dom( f ) -> f [u] = g [ f | (u) ]);
  intros.
  { apply H3; unfold Ordinal_Number; apply AxiomII; split; Ens.
    apply Theorem111 with (x:=dom(f)); auto. }
  assert (forall u : Class, u ∈ dom(h) -> h [u] = g [ h | (u) ]);
```

```
  intros.
  { apply H4; unfold Ordinal_Number; apply AxiomII; split; Ens.
    apply Theorem111 with (x:=dom(h)); auto. }
  generalize (Theorem127 f h g H H1 H5 H0 H2 H6); intro; destruct H7.
  - symmetry; eapply Lemma128''''; eauto.
  - eapply Lemma128''''; eauto.
Qed.

Hint Resolve Theorem128 Theorem128': set.

End A8.

Export A8.
```

定理 128 就是说, 已给 g 可能求得一个在序上唯一的函数 f 使得对于每个序数 $x, f(x) = g(f|x)$, 于是值 $f(x)$ 完全由 g 和 f 在 x 前面的序数处的值所决定. 这个定理的应用就是用 "**超限归纳法来定义一个函数**". 它的证明, 包括定理 127 的证明, 类似于定理 99.

在定理 128 中, 如果 f 的定义域不是 R, 则对于每个使得 f 的定义域 $\leqslant x$ 的序数 x, $g(f) = \mathcal{U}$ 且 $f(x) = \mathcal{U}$. 如果 $g(0) = \mathcal{U}$, 则 $f = 0$.

3.9 非负整数

在本节中定义非负整数, 并且 Peano 公理将作为定理推出. 实数可以利用这些公理由整数和下面两点事实来构造 [45] (本书作者团队已初步完成文献 [45] 中 301 条定理的形式化证明): 整数类是一个集 (定理 138), 同时利用归纳法在整数上定义一个函数是可能的 (这个事实可以作为定理 128 的一个系推出). 此外还需要下面的无限性公理.

无限性公理 VIII (VIII Axiom of infinity) $\exists y$, y 是集 $\Longrightarrow$ $0 \in y \wedge (x \in y \Rightarrow x \cup \{x\} \in y)$.

```
Require Export A_8.

(* A.9 非负整数 *)

Module A9.

Axiom AxiomVIII : ∃ y, Ensemble y /\ ∅ ∈ y /\
  (∀ x, x ∈ y -> (x ∪ [x]) ∈ y).

Hint Resolve AxiomVIII : set.
```

特别地, 0 为一个集, 因为 0 被包含在一个集内.

定义 129 x 是整数 $\Longleftrightarrow x$ 是序 $\wedge E^{-1}$ 良序 x.

```
Definition Integer x : Prop := Ordinal x /\
  WellOrdered (E⁻¹) x.

Hint Unfold Integer : set.
```

定义 130 x 是 y 的一个 E- 末元 $\Longleftrightarrow x$ 是 y 的一个 E^{-1}-首元.

```
Definition LastMember x E y : Prop := FirstMember x (E⁻¹) y.

Hint Unfold LastMember : set.
```

定义 131 $\omega = \{x : x$ 是整数$\}$.

```
Definition ω : Class := \{ λ x, Integer x \}.

Hint Unfold ω : set.
```

定理 132 一个整数的元是一个整数.

```
Theorem Theorem132 : ∀ x y,
  Integer x -> y ∈ x -> Integer y.
Proof.
  intros.
  unfold Integer in H; unfold Integer; destruct H.
  double H; apply Lemma_y with (B := y ∈ x) in H2; auto.
  apply Theorem111 in H2; split; auto.
  unfold WellOrdered in H1; unfold WellOrdered.
  unfold Ordinal in H; destruct H.
  unfold full in H3; apply H3 in H0.
  destruct H1; split; intros.
  - unfold Connect in H1; unfold Connect; intros.
    apply H1; destruct H5; unfold Included in H0.
    apply H0 in H5; apply H0 in H6; split; auto.
  - destruct H5; apply H4; split; auto.
    apply (Theorem28 _ y _); auto.
Qed.

Hint Resolve Theorem132 : set.
```

定理 133 $y \in R \wedge x$ 是 y 的一个 E- 末元 $\Longleftrightarrow y = x + 1$.

```
Theorem Theorem133 : ∀ x y,
```

```
  y ∈ R /\ LastMember x E y -> y = PlusOne x.
Proof.
  intros; destruct H.
  unfold LastMember, FirstMember in H0.
  unfold R in H; apply AxiomII in H; destruct H, H0.
  double H1; add (x ∈ y) H3; apply Theorem111 in H3.
  assert (x ∈ R). { unfold R; apply AxiomII; Ens. }
  apply Theorem123 in H4; unfold FirstMember in H4; destruct H4.
  assert (y ∈ \{ λ z, z ∈ R /\ x ≺ z \}).
  { apply AxiomII; repeat split; auto.
    unfold R; apply AxiomII; split; auto. }
  apply H5 in H6; clear H5; generalize (Theorem113); intros.
  destruct H5; clear H7; apply Theorem107 in H5.
  unfold WellOrdered in H5; destruct H5; clear H7.
  unfold Connect in H5; apply AxiomII in H4; destruct H4, H7.
  clear H8; assert (y ∈ R /\ (PlusOne x) ∈ R).
  { split; auto; unfold R; apply AxiomII; Ens. }
  apply H5 in H8; clear H5; destruct H8; try contradiction.
  destruct H5; auto; unfold Rrelation, E in H5.
  apply AxiomII_P in H5; destruct H5.
  apply H2 in H8; elim H8; unfold Rrelation, Inverse.
  apply AxiomII_P; split; try apply Theorem49; Ens.
  unfold E; apply AxiomII_P; split; try apply Theorem49; Ens.
  unfold PlusOne; apply Theorem4; right.
  unfold Singleton; apply AxiomII; Ens.
Qed.

Hint Resolve Theorem133 : set.
```

定理 134 $x \in \omega \Longleftrightarrow x+1 \in \omega$.

```
Theorem Theorem134 : ∀ x,
  x ∈ ω -> (PlusOne x) ∈ ω.
Proof.
  intros.
  unfold ω in H; apply AxiomII in H; destruct H.
  unfold Integer in H0; destruct H0.
  unfold ω; apply AxiomII; split.
  - unfold PlusOne; apply AxiomIV; split; auto.
    apply Theorem42 in H; auto.
  - unfold Integer; split.
    + assert (x ∈ R). { apply AxiomII; Ens. }
      apply Lemma123 in H2; apply AxiomII in H2; apply H2.
    + unfold WellOrdered in H1; unfold WellOrdered.
      destruct H1; split; intros.
```

```
      { clear H2; unfold Connect in H1; unfold Connect; intros.
        unfold PlusOne in H2; destruct H2; apply Theorem4 in H2.
        apply Theorem4 in H3; destruct H2, H3.
        - apply H1; auto.
        - unfold Singleton in H3; apply AxiomII in H3; destruct H3.
          rewrite <- H4 in H2; try apply Theorem19; Ens.
          right; left; unfold Rrelation, Inverse, E.
          apply AxiomII_P; split; try apply Theorem49; Ens.
          apply AxiomII_P; split; try apply Theorem49; Ens.
        - unfold Singleton in H2; apply AxiomII in H2; destruct H2.
          rewrite <- H4 in H3; try apply Theorem19; Ens.
          left; unfold Rrelation, Inverse, E.
          apply AxiomII_P; split; try apply Theorem49; Ens.
          apply AxiomII_P; split; try apply Theorem49; Ens.
        - unfold Singleton in H2; apply AxiomII in H2; destruct H2.
          unfold Singleton in H3; apply AxiomII in H3; destruct H3.
          right; right; rewrite H4, H5; try apply Theorem19; Ens. }
      { destruct H3; unfold PlusOne in H3.
        generalize (classic (x ∈ y)); intro; destruct H5.
        - exists x; unfold FirstMember; split; intros; auto.
          intro; unfold Rrelation in H7; apply AxiomII_P in H7.
          destruct H7; apply AxiomII_P in H8; destruct H8.
          apply H3 in H6; apply Theorem4 in H6; destruct H6.
          + eapply Theorem102; eauto.
          + apply AxiomII in H6; destruct H6.
            rewrite H10 in H9; try apply Theorem19; Ens.
            apply Theorem101 in H9; auto.
        - apply H2; split; auto; unfold Included; intros; double H6.
          apply H3 in H6; apply Theorem4 in H6; destruct H6; auto.
          apply AxiomII in H6; destruct H6; apply Theorem19 in H.
          rewrite <- H8 in H5; auto; contradiction. }
Qed.

Hint Resolve Theorem134 : set.
```

定理 135 $0 \in \omega$ 并且 $(x \in \omega \Longrightarrow 0 \neq x + 1)$.

```
Theorem Theorem135 : ∀ x, ∅ ∈ ω /\ (x ∈ ω -> ∅ ≠ PlusOne x).
Proof.
  intros; split; intros.
  - unfold ω; apply AxiomII; split.
    + generalize AxiomVIII; intros; destruct H, H, H0; Ens.
    + unfold Integer; split.
      * unfold Ordinal; split.
```

```
      -- unfold Connect; intros; destruct H.
         generalize (Theorem16 u); contradiction.
      -- unfold full; intros.
         generalize (Theorem16 m); contradiction.
    * unfold WellOrdered; split; intros.
      -- unfold Connect; intros; destruct H.
         generalize (Theorem16 u); contradiction.
      -- destruct H; generalize (Theorem26 y); intros.
         absurd (y = ∅); try apply Theorem27; auto.
  - intro; unfold PlusOne in H0; assert (x ∈ ∅).
    { rewrite H0; apply Theorem4; right.
      unfold Singleton; apply AxiomII; split; Ens. }
    generalize (Theorem16 x); intro; contradiction.
Qed.

Hint Resolve Theorem135 : set.
```

也就是说, 0 非整数的后继.

定理 136 x 和 y 均为 ω 的元, 且 $x+1=y+1 \Longrightarrow x=y$.

```
Theorem Theorem136 : ∀ x y,
  x ∈ ω /\ y ∈ ω -> PlusOne x = PlusOne y -> x = y.
Proof.
  intros; destruct H.
  unfold ω in H, H1; apply AxiomII in H.
  apply AxiomII in H1; destruct H, H1.
  unfold Integer in H2, H3; destruct H2, H3.
  assert (x ∈ R /\ y ∈ R).
  { split; unfold R; apply AxiomII; auto. }
  destruct H6; apply Theorem124 in H6.
  apply Theorem124 in H7; rewrite <- H6, <- H7.
  rewrite H0; auto.
Qed.

Hint Resolve Theorem136 : set.
```

下面的定理是**数学归纳法原理**.

定理 137 $(x \subset \omega \wedge 0 \in x \wedge (u \in x \Longrightarrow u+1 \in x)) \Longrightarrow x=\omega$.

```
Corollary Property_ω : Ordinal ω.
Proof.
  unfold Ordinal; split.
  - unfold Connect; intros; destruct H; unfold ω in H, H0.
    apply AxiomII in H; apply AxiomII in H0; destruct H, H0.
```

```
    unfold Integer in H1, H2; destruct H1, H2; add (Ordinal v) H1.
    apply Theorem110 in H1; destruct H1 as [H1|[H1|H1]]; try tauto.
    + left; unfold Rrelation, E; apply AxiomII_P.
      split; auto; apply Theorem49; split; auto.
    + right; left; unfold Rrelation, E; apply AxiomII_P.
      split; auto; apply Theorem49; split; auto.
  - unfold full; intros; unfold Included; intros.
    unfold ω in H; apply AxiomII in H; destruct H.
    apply (Theorem132 _ z) in H1; auto.
    unfold ω; apply AxiomII; Ens.
Qed.

Theorem Theorem137 : ∀ x, x ⊂ ω -> ∅ ∈ x ->
  (∀ u, u ∈ x -> (PlusOne u) ∈ x) -> x = ω.
Proof.
  intros.
  generalize (classic (x = ω)); intros; destruct H2; auto.
  assert (exists y, FirstMember y E (ω ~ x)).
  { assert (WellOrdered E ω).
    { apply Theorem107; apply Property_ω. }
    unfold WellOrdered in H3; destruct H3; apply H4; split.
    - unfold Included; intros; unfold Setminus in H5.
      apply Theorem4' in H5; apply H5.
    - intro; apply Property_∅ in H; apply H in H5.
      symmetry in H5; contradiction. }
  destruct H3 as [y H3]; unfold FirstMember in H3; destruct H3.
  unfold Setminus in H3; apply Theorem4' in H3; destruct H3.
  unfold ω in H3; apply AxiomII in H3; destruct H3; double H6.
  unfold Integer in H7; destruct H7; unfold WellOrdered in H8.
  destruct H8; assert (y ⊂ y /\ y ≠ ∅).
  { split; try unfold Included; auto.
    intro; rewrite H10 in H5; unfold Complement in H5.
    apply AxiomII in H5; destruct H5; contradiction. }
  apply H9 in H10; clear H9; destruct H10 as [u H9].
  assert (u ∈ x).
  { unfold FirstMember in H9; destruct H9; clear H10.
    generalize (classic (u ∈ x)); intros; destruct H10; auto.
    assert (u ∈ (ω ~ x)).
    { unfold Setminus; apply Theorem4'; split.
      - unfold ω; apply AxiomII; split; Ens.
        apply Theorem132 in H9; auto.
      - unfold Complement; apply AxiomII; Ens. }
    apply H4 in H11; elim H11; unfold Rrelation, E.
    apply AxiomII_P; split; try apply Theorem49; Ens. }
  assert (y ∈ R /\ LastMember u E y).
```

```
  { split; auto; unfold R; apply AxiomII; Ens. }
  apply Theorem133 in H11; apply H1 in H10; rewrite <- H11 in H10.
  clear H11; unfold Complement in H5; apply AxiomII in H5.
  destruct H5; unfold NotIn in H11; contradiction.
Qed.

Hint Resolve Theorem137 : set.
```

定理 134—定理 137 均为关于整数的 Peano 公理. 下面的定理推出 ω 是一个集.

定理 138 $\omega \in R$.

```
Theorem Theorem138 : ω ∈ R.
Proof.
  unfold R; apply AxiomII; split; try apply Property_ω.
  generalize AxiomVIII; intros; destruct H, H, H0.
  assert (ω ∩ x = ω).
  { apply Theorem137; intros.
    - unfold Included; intros; apply Theorem4' in H2; apply H2.
    - apply Theorem4'; split; auto; apply Theorem135; auto.
    - apply Theorem4' in H2; destruct H2; apply Theorem134 in H2.
      apply H1 in H3; apply Theorem4'; split; auto. }
  rewrite <- H2; apply Theorem33 with (x:=x); auto.
  unfold Included; intros; apply Theorem4' in H3; apply H3.
Qed.

Hint Resolve Theorem138 : set.
```

常见的**数学归纳法原理**及**第二数学归纳法原理**叙述如下 [91].

定理(数学归纳法原理) 设有一个与正整数 n 有关的命题. 如果: (i) 当 $n=1$ 时命题成立; (ii) 假设 $n=k$ 时命题成立, 则 $n=k+1$ 时命题也成立, 那么这个命题对于一切正整数 n 都成立.

定理(第二数学归纳法原理) 设有一个与正整数 n 有关的命题. 如果: (i) 当 $n=1$ 时命题成立; (ii) 假设命题对于一切小于 k 的正整数来说成立, 则命题对于 k 也成立, 那么这个命题对于一切正整数 n 都成立.

它们的证明是通过**最小数原理**来完成的 [91].

定理(最小数原理) 正整数集的任意一个非空子集 S 必含有一个最小数, 也就是这样一个数 $a \in S$, 对于任意 $x \in S$ 都有 $a \leqslant x$.

补充它们的 Coq 描述及形式化证明如下.

```
Theorem MiniMember_Principle : ∀ S,
```

```
  S ⊂ ω /\ S ≠ ∅ -> ∃ a, a ∈ S /\ (∀ x, x ∈ S -> a ≼ x).
Proof.
  intros; destruct H.
  assert (exists y, FirstMember y E S).
  { assert (WellOrdered E ω).
    { apply Theorem107; apply Property_ω. }
    unfold WellOrdered in H1; destruct H1; apply H2; auto. }
  destruct H1; exists x; unfold FirstMember in H1; destruct H1.
  split; auto; intros; double H3; apply H2 in H4.
  unfold Included in H; apply H in H1; apply H in H3.
  unfold ω in H1, H3; apply AxiomII in H1; apply AxiomII in H3.
  destruct H1, H3; unfold Integer in H5, H6; destruct H5, H6.
  add (Ordinal c) H5; clear H6 H7 H8; apply Theorem110 in H5.
  unfold LessEqual; destruct H5 as [H5|[H5|H5]]; try tauto.
  elim H4; unfold Rrelation, E; apply AxiomII_P; split; auto.
  apply Theorem49; split; Ens.
Qed.

Definition En_S P : Class := \{ λ x, x ∈ ω /\
  ∼ (P x) \}.

Theorem Mathematical_Induction : ∀ (P: Class -> Prop),
  P ∅ -> (∀ k, k ∈ ω /\ P k -> P (PlusOne k)) ->
  (∀ n, n ∈ ω -> P n).
Proof.
  intros.
  generalize (classic ((En_S P) = ∅)); intros; destruct H2.
  - generalize (classic (P n)); intros; destruct H3; auto.
    assert (n ∈ (En_S P)). { apply AxiomII; split; Ens. }
    rewrite H2 in H4; generalize (Theorem16 n); contradiction.
  - assert ((En_S P) ⊂ ω).
    { unfold En_S, Included; intros; apply AxiomII in H3; apply H3. }
    add ((En_S P) <> ∅) H3; clear H2.
    apply MiniMember_Principle in H3; destruct H3 as [h H3], H3.
    unfold En_S in H2; apply AxiomII in H2; destruct H2, H4.
    unfold ω in H4; apply AxiomII in H4; clear H2; destruct H4.
    double H4; unfold Integer in H6; destruct H6.
    unfold WellOrdered in H7; destruct H7.
    assert (h ⊂ h /\ h ≠ ∅).
    { split; try (unfold Included; intros; auto).
      generalize (classic (h = ∅)); intros; destruct H9; auto.
      rewrite H9 in H5; contradiction. }
    apply H8 in H9; clear H8; destruct H9.
    assert (h ∈ R /\ LastMember x E h).
    { split; auto; unfold R; apply AxiomII; split; auto. }
```

```
    apply Theorem133 in H9; unfold PlusOne in H9.
    unfold FirstMember in H8; destruct H8.
    generalize (classic (x ∈ (En_S P))); intros; destruct H11.
    + apply H3 in H11; assert (x ∈ h).
      { rewrite H9; apply Theorem4; right; apply AxiomII; Ens. }
      unfold LessEqual in H11; destruct H11.
      * add (x ∈ h) H11; clear H12.
        generalize (Theorem102 h x); intros; contradiction.
      * rewrite H11 in H12; generalize (Theorem101 x); contradiction.
    + assert (x ∈ (En_S P) <-> (Ensemble x /\ x ∈ ω /\ ~ (P x))).
      { unfold En_S; split; intros.
        - apply AxiomII in H12; apply H12.
        - apply AxiomII; auto. }
      apply Lemma_z in H12; auto; clear H11.
      apply not_and_or in H12; destruct H12.
      * absurd (Ensemble x); Ens.
      * assert (x ∈ ω).
        { unfold ω; apply AxiomII; split; Ens.
          apply Theorem132 in H8; auto. }
        apply not_and_or in H11; destruct H11; try contradiction.
        apply NNPP in H11; add (P x) H12; clear H11.
        apply H0 in H12; unfold PlusOne in H12.
        rewrite <- H9 in H12; contradiction.
Qed.

Theorem The_Second_Mathematical_Induction : ∀ (P: Class -> Prop),
  P ∅ -> (∀ k, k ∈ ω /\ (∀ m, m ≺ k -> P m) -> P k) ->
  (∀ n, n ∈ ω -> P n).
Proof.
  intros; apply H0; split; auto.
  generalize dependent n.
  set (P' := (fun n => (forall j : Class, j ∈ n -> P j))).
  generalize (Mathematical_Induction P'); intro.
  apply H1; red; intros.
  - destruct (Theorem16 j H2).
  - destruct H2; apply H0; split; intros.
    + apply Theorem134 in H2.
      unfold ω in H2; apply AxiomII in H2; destruct H2.
      eapply Theorem132 in H5; eauto.
      unfold ω; apply AxiomII; Ens.
    + apply H4.
      unfold PlusOne in H3; apply AxiomII in H3; destruct H3, H6.
      * unfold ω in H2; apply AxiomII in H2; destruct H2.
        unfold Integer in H7; destruct H7.
        unfold Ordinal in H7; destruct H7.
```

```
        unfold full in H9; eapply H9; eauto.
      * apply AxiomII in H6; destruct H6.
        AssE k; apply Theorem19 in H8; apply H7 in H8.
        subst j; auto.
Qed.

Theorem The_Second_Mathematical_Induction' : ∀ (P: Class -> Prop),
  P ∅ -> (∀ k, k ∈ ω /\ (∀ j, j ∈ k -> P j) -> P k) ->
  (∀ n, n ∈ ω -> P n).
Proof.
  intros.
  generalize (classic ((En_S P) = ∅)); intros; destruct H2.
  - generalize (classic (P n)); intros; destruct H3; auto.
    assert (n ∈ (En_S P)). { apply AxiomII; split; Ens. }
    rewrite H2 in H4; generalize (Theorem16 n); contradiction.
  - assert ((En_S P) ⊂ ω).
    { unfold En_S, Included; intros; apply AxiomII in H3; apply H3. }
    add ((En_S P) <> ∅) H3; clear H2.
    apply MiniMember_Principle in H3; destruct H3 as [h H3], H3.
    unfold En_S in H2; apply AxiomII in H2; destruct H2, H4.
    assert (forall a, a ∈ ω -> a = ∅ \/ exists b, a = (PlusOne b)).
    { apply Mathematical_Induction; auto; intros; right; eauto. }
    destruct H6 with h; auto.
    + rewrite H7 in H5; contradiction.
    + destruct H7; subst h.
      elim H5; apply H0; split; auto; intros.
      generalize (classic (P j)); intros; destruct H8; auto.
      assert (j ∈ (En_S P)).
      { apply AxiomII; repeat split; Ens.
        unfold ω in H4; apply AxiomII in H4; clear H2; destruct H4.
        apply AxiomII; split; Ens.
        eapply Theorem132; eauto. }
      apply H3 in H9; destruct H9.
      * add ((PlusOne x) ∈ j) H7; clear H9.
        destruct (Theorem102 _ _ H7).
      * subst j; destruct (Theorem101 _ H7).
Qed.
```

在上面的代码中, 我们给出了第二数学归纳法原理的两种不同的证明方法：一种是利用数学归纳法原理；另一种直接应用最小数原理.

事实上, 定理 137 与常用的数学归纳法原理是等价的, 其证明代码如下.

```
(* Mathematical Induction Theorem137 *)

Theorem Mathematical_Induction_Theorem137 : ∀ (P: Class -> Prop),
```

```
  (P ∅ -> (∀ k, k ∈ ω /\ P k -> P (PlusOne k)) ->
  (∀ n, n ∈ ω -> P n)) <->
  (\{ λ x, x ∈ ω /\ P x \} ⊂ ω -> ∅ ∈ \{λ x, x ∈ ω /\ P x\} -> (∀ u,
  u ∈ \{ λ x, x ∈ ω /\ P x \} -> (PlusOne u) ∈ \{ λ x, x ∈ ω /\ P x
   \}) -> \{ λ x, x ∈ ω /\ P x \} = ω).
Proof.
  split; intros.
  - apply AxiomII in H1; destruct H1, H3; apply AxiomI; split; intros.
    + apply H0 in H5; auto.
    + apply H with (n:= z) in H4; auto; try apply AxiomII; Ens; intros.
      destruct H6; assert (k ∈ \{λ x, x∈ ω /\ P x\}).
      { apply AxiomII; Ens. }
      apply H2 in H8; apply AxiomII in H8; apply H8.
  - assert (\{ λ x, x ∈ ω /\ P x \} ⊂ ω).
    { unfold Included; intros; apply AxiomII in H3; apply H3. }
    assert (∅ ∈ \{ λ x, x ∈ ω /\ P x \}).
    { apply AxiomII; generalize (Theorem135 n); intros.
      destruct H4 as [H4 _]; Ens. }
    assert ((forall u, u ∈ \{ λ x, x ∈ ω /\ P x \} ->
     (PlusOne u) ∈ \{ λ x, x ∈ ω /\ P x \})).
    { intros; apply AxiomII in H5; destruct H5, H6; double H6.
      apply Theorem134 in H8; apply AxiomII; repeat split; Ens. }
    apply H in H5; auto; rewrite <- H5 in H2; clear H H3 H4 H5.
    apply AxiomII in H2; apply H2.
Qed.

End A9.

Export A9.
```

3.10 选 择 公 理

定义 139 c 是选择函数 $\iff$ c 是函数 $\wedge$ ($\forall x \in c$ 的定义域, $c(x) \in x$).

```
Require Export A_9.

(* A.10 选择公理 *)

Module A10.

Definition ChoiceFunction c : Prop :=
  Function c /\ (∀ x, x ∈ dom(c) -> c[x] ∈ x).

Hint Unfold ChoiceFunction : set.
```

下面是**Zermelo 假定**的一个强化形式, 或称为**选择公理**.

选择公理 IX (IX Axiom of choice) ∃ 选择函数 c, 它的定义域是 $\mathcal{U} \sim \{0\}$.

```
Axiom AxiomIX : ∃ c, ChoiceFunction c /\ dom(c) = 𝒰 ∼ [∅].
```

```
Hint Resolve AxiomIX : set.
```

函数 c 从每个非空集中选取一个元.

定理 140 x 是集 $\Longrightarrow$ ($\exists f$, f 是 1-1 函数, f 的值域 $= x$, f 的定义域是一个序数).

```
Lemma Ex_Lemma140 : ∀ x c,
  Ensemble x -> ChoiceFunction c ->
  (∃ g, ∀ h, Ensemble h -> g[h] = c[x ∼ ran(h)]).
Proof.
  intros.
  unfold ChoiceFunction in H0; destruct H0.
  exists (\{\ λ u v, v = c [x ∼ ran(u)] \}\); intros.
  apply AxiomI; split; intros.
  - apply AxiomII; split; Ens; intros.
    apply AxiomII in H3; destruct H3.
    apply H5; clear H5; apply AxiomII; split; Ens.
    apply AxiomII_P; split; try apply Theorem49; Ens.
    apply AxiomII in H4; destruct H4.
    rewrite Theorem70 in H5; auto.
    apply AxiomII_P in H5; apply H5.
  - apply AxiomII; split; Ens; intros.
    apply AxiomII in H4; destruct H4.
    apply AxiomII_P in H5; destruct H5.
    rewrite H6; auto.
Qed.

Lemma Property_Relation : ∀ f,
  dom(f⁻¹) = ran(f) /\ ran(f⁻¹) = dom(f).
Proof.
  intros; unfold Domain, Range; split.
  - apply AxiomI; split; intros.
    + apply AxiomII in H; destruct H, H0; apply AxiomII_P in H0.
      destruct H0; apply AxiomII; split; Ens.
    + apply AxiomII in H; destruct H, H0; apply AxiomII; split; auto.
      exists x; apply AxiomII_P; split; auto; apply Theorem49.
      AssE [x,z]; apply Theorem49 in H1; destruct H1; auto.
  - apply AxiomI; split; intros.
    + apply AxiomII in H; destruct H, H0; apply AxiomII_P in H0.
      destruct H0; apply AxiomII; split; Ens.
```

```
    + apply AxiomII in H; destruct H, H0; apply AxiomII; split; auto.
      exists x; apply AxiomII_P; split; auto; apply Theorem49.
      AssE [z,x]; apply Theorem49 in H1; destruct H1; auto.
Qed.

Lemma Lemma140 : ∀ f g y,
  y ∈ dom(f) -> f [y] = g [f | (y)] -> Ensemble (f | (y)).
Proof.
  intros.
  generalize (classic ((f|(y)) ∈ dom(g))); intros; destruct H1; Ens.
  apply Theorem69 in H1; rewrite H1 in H0; clear H1.
  apply Theorem69 in H; rewrite H0 in *.
  generalize (Theorem101 𝓤); intros; contradiction.
Qed.

Theorem Theorem140 : ∀ x,
  Ensemble x -> ∃ f, Function1_1 f /\ ran(f) = x /\
  Ordinal_Number dom(f).
Proof.
  intros.
  generalize AxiomIX; intros; destruct H0 as [c H0], H0.
  double H0; apply (Ex_Lemma140 x_) in H2; auto; destruct H2 as [g H2].
  generalize (Theorem128 g); intros; destruct H3 as [f H3], H3, H4.
  unfold ChoiceFunction in H0; destruct H0; exists f.
  assert (Function1_1 f).
  { unfold Function1_1; split; auto.
    unfold Function; split; intros.
    - unfold Relation; intros; PP H7 a b; Ens.
    - unfold Inverse in H7; destruct H7.
      apply AxiomII_P in H7; apply AxiomII_P in H8; destruct H7, H8.
      clear H7 H8; double H9; apply Property_dom in H8.
      double H10; apply Property_dom in H10.
      generalize (classic (y = z)); intros; destruct H11; auto.
      assert (Ordinal y /\ Ordinal z).
      { split; apply (Theorem111 dom(f) _); auto. }
      elim H12; intros; apply Theorem110 in H12.
      assert (Ordinal_Number y /\ Ordinal_Number z).
      { unfold Ordinal_Number, R; split; apply AxiomII; Ens. }
      clear H13 H14; destruct H15; apply H5 in H13; apply H5 in H14.
      rewrite H2 in H13, H14; try apply (Lemma140 _ g _); auto.
      clear H2 H5; apply Property_Value in H8; auto.
      apply Property_Value in H10; auto.
      unfold Function in H3; destruct H3.
      add ([y,f[y]] ∈ f) H7; add ([z,f[z]] ∈ f) H9.
      apply H3 in H7; apply H3 in H9; rewrite H9 in H7; clear H9.
```

```
    double H8; double H10; apply Property_ran in H8.
    apply Property_ran in H10; destruct H12.
    + assert (f[z] ∈ ran(f|(z))).
      { rewrite H7; unfold Range; apply AxiomII; split; Ens.
        exists y; unfold Restriction; apply Theorem4'; split; auto.
        unfold Cartesian; apply AxiomII_P; split; Ens.
        split; auto; apply Theorem19; Ens. }
      assert ((x ~ ran(f|(z))) ∈ dom(c)).
      { generalize (classic ((x ~ ran(f|(z))) ∈ dom(c))); intros.
        destruct H16; auto; apply Theorem69 in H16; auto.
        rewrite H16 in H14; rewrite H14 in H10; AssE 𝓤.
        generalize Theorem39; intros; contradiction. }
      apply H6 in H16; unfold Setminus at 2 in H16.
      rewrite <- H14 in H16; apply Theorem4' in H16; destruct H16.
      unfold Complement in H17; apply AxiomII in H17; destruct H17.
      unfold NotIn in H18; contradiction.
    + destruct H12; try contradiction.
      assert (f[y] ∈ ran(f|(y))).
      { rewrite <- H7; unfold Range; apply AxiomII; split; Ens.
        exists z; unfold Restriction; apply Theorem4'; split; auto.
        unfold Cartesian; apply AxiomII_P; split; Ens.
        split; auto; apply Theorem19; Ens. }
      assert ((x ~ ran(f|(y))) ∈ dom(c)).
      { generalize (classic ((x ~ ran(f|(y))) ∈ dom(c))); intros.
        destruct H16; auto; apply Theorem69 in H16; auto.
        rewrite H16 in H13; rewrite H13 in H8; AssE 𝓤.
        generalize Theorem39; intros; contradiction. }
      apply H6 in H16; unfold Setminus at 2 in H16.
      rewrite <- H13 in H16; apply Theorem4' in H16; destruct H16.
      unfold Complement in H17; apply AxiomII in H17; destruct H17.
      unfold NotIn in H18; contradiction. }
split; auto; assert (ran(f) ⊂ x).
{ unfold Included; intros; unfold Range in H8; apply AxiomII in H8.
  destruct H8, H9; double H9; apply Property_dom in H10.
  assert (Ordinal_Number x0).
  { unfold Ordinal_Number, R; apply AxiomII; split; Ens.
    apply (Theorem111 dom(f) _); split; auto. }
  apply H5 in H11; rewrite H2 in H11; try apply (Lemma140_g_); auto.
  apply Property_Value in H10; auto; destruct H3.
  add ([x0,f[x0]]∈f) H9; apply H12 in H9; rewrite <- H9 in H11.
  assert ((x ~ ran(f|(x0))) ∈ dom(c)).
  { generalize (classic ((x ~ ran(f|(x0))) ∈ dom(c))); intros.
    destruct H13; auto; apply Theorem69 in H13; auto.
    rewrite H13 in H11; rewrite H11 in H9; rewrite <- H9 in H10.
    clear H9 H11 H13; apply Property_ran in H10; AssE 𝓤.
```

```
    generalize Theorem39; intros; contradiction. }
  apply H6 in H13; rewrite <- H11 in H13.
  unfold Setminus in H13; apply Theorem4' in H13; apply H13. }
assert (Ensemble dom(f)).
{ unfold Function1_1 in H7; destruct H7 as [H9 H7]; clear H9.
  generalize (Property_Relation f); intros;
  destruct H9; rewrite <- H9 in H8.
  rewrite <- H10; apply AxiomV; apply Theorem33 in H8; auto. }
assert (Ordinal_Number dom(f)).
{ unfold Ordinal_Number; apply AxiomII; split; auto. }
split; auto; apply H5 in H10.
assert (f|(dom(f)) = f).
{ unfold Restriction; apply AxiomI; split; intros.
  - apply AxiomII in H11; apply H11.
  - apply AxiomII; repeat split;Ens;unfold Function, Relation in H3.
    double H11; apply H3 in H12; destruct H12 as [a [b H12]].
    rewrite H12 in *; clear H12; apply AxiomII_P; repeat split; Ens.
    + apply Property_dom in H11; auto.
    + apply Property_ran in H11; apply Theorem19; Ens. }
rewrite H11 in *; clear H11.
rewrite H2 in H10; try apply Theorem75; auto.
generalize (Theorem101 dom(f)); intros.
apply Theorem69 in H11; auto; rewrite H10 in H11.
generalize (classic ((x ~ ran(f)) ∈ dom(c))); intros; destruct H12.
- apply Theorem69 in H12; auto; rewrite H11 in H12.
  generalize (Theorem101 𝒰); intros; contradiction.
- rewrite H1 in H12; unfold Setminus at 2 in H12.
  assert ((x ~ ran(f)) ∈ (𝒰 ∩ ¬ [∅]) <-> (x ~ ran(f)) ∈ 𝒰 /\
          (x ~ ran(f)) ∈ ¬ [∅]).
  { split; intros; try apply Theorem4'; auto. }
  apply Lemma_z in H13; auto; clear H12.
  assert (Ensemble (x ~ ran(f))).
  { apply (Theorem33 x _); auto; unfold Included.
    intros; apply AxiomII in H12; apply H12. }
  apply not_and_or in H13; destruct H13.
  + elim H13; apply Theorem19; auto.
  + assert ((x ~ ran(f)) ∈ ¬ [∅] <-> Ensemble (x ~ ran(f)) /\
            (x ~ ran(f)) ∉ [∅]).
    { split; intros; try apply AxiomII; auto.
      apply AxiomII in H14; apply H14. }
    apply Lemma_z in H14; auto; clear H13.
    apply not_and_or in H14; destruct H14; try contradiction.
    unfold NotIn in H13; apply NNPP in H13.
    unfold Singleton in H13; apply AxiomII in H13; destruct H13.
    generalize AxiomVIII; intros; destruct H15, H15, H16.
```

```
    AssE ∅; clear H15 H16 H17; apply Theorem19 in H18.
    apply H14 in H18; symmetry; apply -> Property_∅ in H18; auto.
Qed.

Hint Resolve Theorem140 : set.
```

定义 141 n 是套 $\Longleftrightarrow (x \in n \wedge y \in n \Longrightarrow\ x \subset y \vee y \subset x)$.

```
Definition Nest n : Prop :=
  ∀ x y, x ∈ n /\ y ∈ n -> x ⊂ y \/ y ⊂ x.

Hint Unfold Nest : set.
```

定理 142 (n 是套 $\wedge$ n 的每一个元是套) $\Longrightarrow \bigcup n$ 是套.

```
Theorem Theorem142 : ∀ n,
  Nest n /\ (∀ m, m ∈ n -> Nest m) -> Nest (⋃ n).
Proof.
  intros; destruct H.
  unfold Nest; intros; destruct H1.
  unfold Element_U in H1, H2; apply AxiomII in H1.
  apply AxiomII in H2; destruct H1, H2, H3, H4, H3, H4.
  unfold Nest in H; assert (x0 ∈ n /\ x1 ∈ n). { Ens. }
  apply H0 in H5; apply H0 in H6; apply H in H7; destruct H7.
  - unfold Included in H7; apply H7 in H3; clear H7.
    unfold Nest in H6; apply H6; auto.
  - unfold Included in H7; apply H7 in H4; clear H7.
    unfold Nest in H5; apply H5; auto.
Qed.

Hint Resolve Theorem142 : set.
```

下面的定理是**Hausdorff 极大原则**, 它断言在任何集中极大套的存在性. 这个证明与定理 140 的证明有密切关系.

定理 143 x 是套 $\Longrightarrow (\exists n, n$ 是套$, n \subset x, m$ 是套$, m \subset x, n \subset m \Longrightarrow m = n)$.

```
Definition En_c x h : Class :=
  \{ λ m, Nest m /\ m ⊂ x /\ (∀ p, p ∈ ran(h) -> p ⊂ m /\ p <> m) \}.

Lemma Ex_Lemma143 : ∀ x c,
  Ensemble x -> ChoiceFunction c ->
  (∃ g, ∀ h, Ensemble h -> g[h] = c[En_c x h]).
Proof.
  intros.
  unfold ChoiceFunction in H0; destruct H0.
```

```
  exists (\{\ λ u v, v = c[En_c x u] \}\); intros.
  apply AxiomI; split; intros.
  - apply AxiomII; split; Ens; intros.
    apply AxiomII in H3; destruct H3.
    apply H5; clear H5; apply AxiomII; split; Ens.
    apply AxiomII_P; split; try apply Theorem49; Ens.
    apply AxiomII in H4; destruct H4.
    rewrite Theorem70 in H5; auto.
    apply AxiomII_P in H5; apply H5.
  - apply AxiomII; split; Ens; intros.
    apply AxiomII in H4; destruct H4.
    apply AxiomII_P in H5; destruct H5.
    rewrite H6; auto.
Qed.

Lemma Lemma143 : ∀ f z,
  Function f -> z ∈ (⋃ ran(f)) -> ∃ x, z ∈ f [x] /\ x ∈ dom(f).
Proof.
  intros.
  unfold Element_U in H0; apply AxiomII in H0.
  destruct H0, H1 as [y H1], H1; unfold Range in H2.
  apply AxiomII in H2; destruct H2, H3 as [x H3].
  double H3; rewrite Theorem70 in H4; auto.
  apply AxiomII_P in H4; destruct H4; rewrite H5 in *.
  apply Property_dom in H3; Ens.
Qed.

Theorem Theorem143 : ∀ x,
  Ensemble x -> ∃ n, (Nest n /\ n ⊂ x) /\
  (∀ m, Nest m -> m ⊂ x /\ n ⊂ m -> m = n).
Proof.
  intros.
  generalize AxiomIX; intros; destruct H0 as [c H0], H0.
  double H0; apply (Ex_Lemma143 x_) in H2; auto; destruct H2 as [g H2].
  generalize (Theorem128 g); intros; destruct H3 as [f H3], H3, H4.
  unfold ChoiceFunction in H0; destruct H0.
  (* If u ∈ dom(f), then f[u] is a nest of x.*)
  assert (∀ u, u ∈ dom(f) -> Nest f[u] /\ f[u] ⊂ x /\
        (∀ p, p ∈ ran(f|(u)) -> p ⊊ f[u])).
  { intros; assert (Ordinal_Number u).
    { unfold Ordinal_Number, R; apply AxiomII.
      split; try apply (Theorem111 dom(f) _); Ens. }
    apply H5 in H8; rewrite H2 in H8; try apply (Lemma140 _ g _); auto.
    assert (En_c x (f|(u)) ∈ dom(c)).
    { generalize (classic (En_c x (f|(u)) ∈ dom(c))); intros.
```

```
    destruct H9; auto; apply Theorem69 in H9; auto.
    rewrite H9 in H8; clear H9; apply Property_Value in H7; auto.
    apply Property_ran in H7; rewrite H8 in H7; AssE 𝓤.
    generalize Theorem39; intros; contradiction. }
  apply H6 in H9; rewrite <- H8 in H9; clear H8.
  apply AxiomII in H9; apply H9. }
(* If u and v are members of dom(f), and u<v, then f[u] ⊊ f[v]. *)
assert (∀ u v, u ∈ dom(f) -> v ∈ dom(f) -> u ∈ v -> f[u] ⊊ f[v]).
{ intros; apply H7 in H9; clear H7; destruct H9, H9.
  apply H11; unfold Range; apply AxiomII.
  apply Property_Value in H8; auto; double H8.
  apply Property_ran in H12; split; Ens; exists u.
  unfold Restriction; apply Theorem4'; split; auto.
  unfold Cartesian; apply AxiomII_P; split; Ens.
  split; try apply Theorem19; Ens. }
exists (⋃ ran(f)); intros; split; intros; try split.
- unfold Nest; intros z0 z1 H9; destruct H9.
  apply Lemma143 in H9; apply Lemma143 in H10; auto.
  destruct H9, H10, H9, H10.
  assert (Ordinal x0 /\ Ordinal x1).
  { split; apply (Theorem111 dom(f) _); auto. }
  apply Theorem110 in H13; destruct H13 as [H13|[H13|H13]].
  + apply H8 in H13; auto; unfold Included in H13.
    destruct H13; apply H13 in H9; clear H13 H14.
    apply H7 in H12; destruct H12; clear H13.
    unfold Nest in H12; apply H12; auto.
  + apply H8 in H13; auto; unfold Included in H13.
    destruct H13; apply H13 in H10; clear H13 H14.
    apply H7 in H11; destruct H11; clear H13.
    unfold Nest in H11; apply H11; auto.
  + rewrite H13 in H9; apply H7 in H12; destruct H12.
    unfold Nest in H12; apply H12; auto.
- unfold Included; intros; apply Lemma143 in H9; auto.
  destruct H9, H9; apply H7 in H10; destruct H10, H11.
  unfold Included in H11; apply H11; auto.
- destruct H10; generalize (Theorem101 dom(f)); intros.
  apply Theorem69 in H12.
  assert (Function f⁻¹).
  { unfold Function; split; intros.
    - unfold Relation; intros; PP H13 a b; Ens.
    - destruct H13; unfold Inverse in H13, H14.
      apply AxiomII_P in H13; apply AxiomII_P in H14.
      destruct H13, H14; double H15; double H16.
      rewrite Theorem70 in H17, H18; auto.
      apply AxiomII_P in H17; apply AxiomII_P in H18.
```

```
    destruct H17, H18; rewrite H19 in H20.
    clear H17 H18 H19; apply Property_dom in H15.
    apply Property_dom in H16.
    assert (Ordinal y /\ Ordinal z).
    { split; apply (Theorem111 dom(f) _); auto. }
    apply Theorem110 in H17; destruct H17.
    + apply H8 in H17; auto; unfold Included in H17.
      destruct H17; contradiction.
    + destruct H17; auto; apply H8 in H17; auto.
      destruct H17; symmetry in H20; contradiction. }
assert (Ensemble dom(f)).
{ generalize (Property_Relation f); intros; destruct H14.
  rewrite <- H15; apply AxiomV; auto; rewrite H14.
  apply (Theorem33 pow(x) _); try (apply Theorem38 in H; auto).
  unfold Included; intros; unfold Range in H16;
  apply AxiomII in H16.
  destruct H16, H17; double H17; rewrite Theorem70 in H18; auto.
  apply AxiomII_P in H18; destruct H18;
  rewrite H19 in *; clear H18 H19.
  apply Property_dom in H17; apply H7 in H17; destruct H17, H18.
  unfold PowerSet; apply AxiomII; split; auto. }
assert (Ordinal_Number dom(f)).
{ unfold Ordinal_Number, R; apply AxiomII; split; auto. }
apply H5 in H15; assert (f|(dom(f)) = f).
{ unfold Restriction; apply AxiomI; split; intros.
  - apply AxiomII in H16; apply H16.
  - apply AxiomII; repeat split; Ens;
    unfold Function, Relation in H3.
    double H16; apply H3 in H17; destruct H17 as [a [b H17]].
    rewrite H17 in*; clear H17; apply AxiomII_P; repeat split; Ens.
    + apply Property_dom in H16; auto.
    + apply Property_ran in H16; apply Theorem19; Ens. }
rewrite H16 in *; rewrite H15 in H12; clear H15 H16.
rewrite H2 in H12; try apply Theorem75; auto.
generalize (classic (En_c x f ∈ dom(c))); intros; destruct H15.
+ apply Theorem69 in H15; auto; rewrite H12 in H15.
  generalize (Theorem101 𝒰); intros; contradiction.
+ rewrite H1 in H15; unfold Setminus in H15.
  assert (En_c x f ∈ (𝒰 ∩ ¬ [∅]) <-> En_c x f ∈ 𝒰 /\
  En_c x f ∈ ¬ [∅]).
  { split; intros; try apply Theorem4'; auto. }
  apply Lemma_z in H16; auto; clear H15.
  assert (Ensemble (En_c x f)).
  { apply (Theorem33 pow(x) _); try (apply Theorem38 in H; auto).
    unfold Included; intros; unfold En_c in H15.
```

```
      apply AxiomII in H15; destruct H15, H17, H18.
      unfold PowerSet; apply AxiomII; split; auto. }
    apply not_and_or in H16; destruct H16.
    * elim H16; apply Theorem19; auto.
    * assert (En_c x f ∈ ¬ [∅] <-> Ensemble (En_c x f) /\
      En_c x f ∉ [∅]).
      { split; intros; try apply AxiomII; auto.
        apply AxiomII in H17; apply H17. }
      apply Lemma_z in H17; auto; clear H16.
      apply not_and_or in H17; destruct H17; try contradiction.
      unfold NotIn in H16; apply NNPP in H16.
      unfold Singleton in H16; apply AxiomII in H16; destruct H16.
      generalize AxiomVIII; intros; destruct H18, H18, H19.
      AssE ∅; clear H18 H19 H20; apply Theorem19 in H21.
      apply H17 in H21; clear H17.
      generalize (classic (m = ⋃ ran( f))); intros;
      destruct H17; auto.
      assert (m ∈ (En_c x f)).
      { unfold En_c; apply AxiomII; repeat split; auto.
        - apply (Theorem33 x _); auto.
        - unfold Included; intros; apply H11.
          unfold Element_U; apply AxiomII; split; Ens.
        - intro; rewrite H19 in H18; clear H19.
          elim H17; apply Theorem27; split; auto.
          unfold Included; intros; apply AxiomII; Ens. }
      rewrite H21 in H18; generalize (Theorem16 m); contradiction.
Qed.

Hint Resolve Theorem143 : set.

End A10.

Export A10.
```

3.11　基　　数

定义 144　$x \approx y \iff (\exists\, f,\ f$ 是 1-1 函数, f 的定义域 $= x$, f 的值域 $= y)$.

```
Require Export A_10.

(* A.11 基数 *)

Module A11.

Definition Equivalent x y : Prop :=
```

```
  ∃ f, Function1_1 f /\ dom(f) = x /\ ran(f) = y.

Notation "x ≈ y" := (Equivalent x y) (at level 70).

Hint Unfold Equivalent : set.
```

如果 $x \approx y$, 则称 x “**等势于**” y, 或者 x 与 y 是 “**等势的**”.

定理 145　$x \approx x$.

```
Theorem Theorem145 : ∀ x, x ≈ x.
Proof.
  intros.
  unfold Equivalent.
  exists (\{\ λ u v, u ∈ x /\ u = v \}\); split.
  - unfold Function1_1; split.
    + unfold Function; split; intros.
      * unfold Relation; intros; PP H a b; Ens.
      * destruct H; apply AxiomII_P in H.
        apply AxiomII_P in H0; destruct H, H0, H1, H2.
        rewrite <- H3, <- H4; auto.
    + unfold Function; split; intros.
      * unfold Relation; intros; PP H a b; Ens.
      * unfold Inverse in H; destruct H; apply AxiomII_P in H.
        apply AxiomII_P in H0; destruct H, H0.
        apply AxiomII_P in H1; apply AxiomII_P in H2.
        destruct H1, H2, H3, H4; rewrite H5, H6; auto.
   - split.
     + apply AxiomI; split; intros.
       * unfold Domain in H; apply AxiomII in H; destruct H, H0.
         apply AxiomII_P in H0; apply H0.
       * unfold Domain; apply AxiomII; split; Ens.
         exists z; apply AxiomII_P; repeat split; auto.
         apply Theorem49; split; Ens.
     + apply AxiomI; split; intros.
       * unfold Range in H; apply AxiomII in H; destruct H, H0.
         apply AxiomII_P in H0; destruct H0, H1.
         rewrite H2 in H1; auto.
       * unfold Range; apply AxiomII; split; Ens.
         exists z; apply AxiomII_P; repeat split; auto.
         apply Theorem49; split; Ens.
Qed.

Hint Resolve Theorem145 : set.
```

定理 146　$x \approx y \implies y \approx x$.

```
Theorem Theorem146 : ∀ x y, x ≈ y -> y ≈ x.
Proof.
  intros.
  unfold Equivalent in H; destruct H as [f H], H, H0.
  unfold Equivalent; exists f⁻¹; split.
  - unfold Function1_1 in H; destruct H.
    unfold Function1_1; split; try rewrite Theorem61;try apply H; auto.
  - unfold Inverse; split.
    + unfold Domain; apply AxiomI; split; intros.
      * apply AxiomII in H2; destruct H2, H3.
        apply AxiomII_P in H3; destruct H3.
        apply Property_ran in H4; rewrite H1 in H4; auto.
      * apply AxiomII; split; Ens.
        rewrite <- H1 in H2; unfold Range in H2.
        apply AxiomII in H2; destruct H2, H3.
        exists (x0); apply AxiomII_P; split; auto.
        apply Theorem49; AssE ([x0,z]).
        apply Theorem49 in H4; destruct H4; Ens.
    + unfold Range; apply AxiomI; split; intros.
      * apply AxiomII in H2; destruct H2, H3.
        apply AxiomII_P in H3; destruct H3.
        apply Property_dom in H4; rewrite H0 in H4; auto.
      * apply AxiomII; split; Ens.
        rewrite <- H0 in H2; unfold Domain in H2.
        apply AxiomII in H2; destruct H2, H3.
        exists (x0); apply AxiomII_P; split; auto.
        apply Theorem49; AssE ([z,x0]).
        apply Theorem49 in H4; destruct H4; Ens.
Qed.

Hint Resolve Theorem146 : set.
```

定理 147　$x \approx y \wedge y \approx z \implies x \approx z.$

```
Theorem Theorem147 : ∀ x y z,
  x ≈ y -> y ≈ z -> x ≈ z.
Proof.
  intros.
  unfold Equivalent in H, H0; unfold Equivalent.
  destruct H as [f1 H], H0 as [f2 H0], H, H0, H1, H2.
  exists (\{\ λ u v, ∃ w, [u,w] ∈ f1 /\ [w,v] ∈ f2\}\); split.
  - unfold Function1_1; unfold Function1_1 in H, H0.
    destruct H, H0; split.
    + unfold Function; split; intros.
      * unfold Relation; intros; PP H7 a b; Ens.
```

```
    * destruct H7; apply AxiomII_P in H7; destruct H7, H9.
      apply AxiomII_P in H8; destruct H8, H10; clear H7 H8.
      unfold Function in H, H0; destruct H9, H10, H, H0.
      add ([x0,x2] ∈ f1) H7; apply H11 in H7; rewrite H7 in H8.
      add ([x2,z0] ∈ f2) H8; apply H12 in H8; auto.
  + unfold Function; split; intros.
    * unfold Relation; intros; PP H7 a b; Ens.
    * unfold Inverse in H7; destruct H7; apply AxiomII_P in H7.
      apply AxiomII_P in H8; destruct H7, H8; clear H7 H8.
      apply AxiomII_P in H9; destruct H9, H8.
      apply AxiomII_P in H10; destruct H10, H10; clear H7 H9.
      unfold Function in H5, H6; destruct H8, H10, H5, H6.
      assert ([x0,x1] ∈ f2⁻¹ /\ [x0,x2] ∈ f2⁻¹).
      { unfold Inverse; split.
        - apply AxiomII_P; split; auto; AssE [x1,x0].
          apply Theorem49 in H13; destruct H13.
          apply Theorem49; split; auto.
        - apply AxiomII_P; split; auto; AssE [x2,x0].
          apply Theorem49 in H13; destruct H13.
          apply Theorem49; split; auto. }
      apply H12 in H13; rewrite H13 in H7; clear H8 H10 H12 H13.
      assert ([x2,y0] ∈ f1⁻¹ /\ [x2,z0] ∈ f1⁻¹).
      { unfold Inverse; split.
        - apply AxiomII_P; split; auto; AssE [y0,x2].
          apply Theorem49 in H8; destruct H8.
          apply Theorem49; split; auto.
        - apply AxiomII_P; split; auto; AssE [z0,x2].
          apply Theorem49 in H8; destruct H8.
          apply Theorem49; split; auto. }
      apply H11 in H8; auto.
- rewrite <- H1, <- H4; split.
  + apply AxiomI; split; intros.
    * apply AxiomII in H5; destruct H5, H6.
      apply AxiomII_P in H6; destruct H6, H7, H7.
      apply Property_dom in H7; auto.
    * apply AxiomII; split; Ens; apply AxiomII in H5.
      destruct H5, H6; double H6; apply Property_ran in H7.
      rewrite H3 in H7; rewrite <- H2 in H7; apply AxiomII in H7.
      destruct H7, H8; exists x1; apply AxiomII_P; split; Ens.
      AssE [z0,x0]; AssE [x0,x1]; apply Theorem49 in H9.
      apply Theorem49 in H10; destruct H9, H10.
      apply Theorem49; split; auto.
  + apply AxiomI; split; intros.
    * apply AxiomII in H5; destruct H5, H6.
      apply AxiomII_P in H6; destruct H6, H7, H7.
```

```
      apply Property_ran in H8; auto.
    * apply AxiomII; split; Ens; apply AxiomII in H5.
      destruct H5, H6; double H6; apply Property_dom in H7.
      rewrite H2 in H7; rewrite <- H3 in H7; apply AxiomII in H7.
      destruct H7, H8; exists x1; apply AxiomII_P; split; Ens.
      AssE [x0,z0]; AssE [x1,x0]; apply Theorem49 in H9.
      apply Theorem49 in H10; destruct H9, H10.
      apply Theorem49; split; auto.
Qed.

Hint Resolve Theorem147 : set.
```

定义 148 x 是基数 $\Longleftrightarrow$ x 是序数 $\wedge$ $((\forall y, y \in R, y < x) \implies \sim (x \approx y))$.

```
Definition Cardinal_Number x : Prop :=
  Ordinal_Number x /\ (∀ y, y ∈ R -> y ≺ x -> ∼ (x ≈ y)).

Hint Unfold Cardinal_Number : set.
```

也就是说, 一个**基数**是一个序数, 而它不等势于任何较小的序数.

定义 149 $C = \{x : x$ 是基数$\}$.

```
Definition C : Class := \{ λ x, Cardinal_Number x \}.

Hint Unfold C : set.
```

定理 150 E 良序 C.

```
Theorem Theorem150 : WellOrdered E C.
Proof.
  intros.
  unfold WellOrdered; split; intros.
  - unfold Connect; intros; destruct H; unfold C in H, H0.
    apply AxiomII in H; apply AxiomII in H0; destruct H, H0.
    unfold Cardinal_Number in H1, H2; destruct H1, H2; clear H3 H4.
    unfold Ordinal_Number, R in H1, H2; apply AxiomII in H1.
    apply AxiomII in H2; destruct H1, H2; add (Ordinal v) H3.
    clear H1 H2 H4; apply Theorem110 in H3; destruct H3.
    + left; unfold Rrelation, E; apply AxiomII_P.
      split; try apply Theorem49; auto.
    + destruct H1; auto; right; left; unfold Rrelation, E.
      apply AxiomII_P; split; try apply Theorem49; auto.
  - destruct H; assert (y ⊂ R).
    { unfold Included; intros; unfold Included in H.
      apply H in H1; unfold C in H1; apply AxiomII in H1.
```

```
    destruct H1; unfold Cardinal_Number in H2; destruct H2.
    unfold Ordinal_Number in H2; auto. }
  add (y ≠ ∅) H1; apply Lemma121 in H1; Ens.
Qed.

Hint Resolve Theorem150 : set.
```

定义 151 $P=\{(x,y):x\approx y\wedge y\in C\}$.

```
Definition P : Class := \{\ λ x y, x ≈ y /\ y ∈ C \}\.

Hint Unfold P : set.
```

类 P 是由所有使得 x 是一个集且 y 是等势于 x 的基数之偶对 (x,y) 所组成的. 对于每一个集 x, 基数 $P(x)$ 是 "x **的势**", 或者 "x **的基数**".

定理 152 P 是函数 $\wedge$ P 的定义域 $=\mathcal{U}$ $\wedge$ P 的值域 $=C$.

```
Theorem Theorem152 : Function P /\ dom(P) = 𝒰 /\ ran(P) = C.
Proof.
  unfold P; repeat split; intros.
  - unfold Relation; intros; PP H a b; Ens.
  - destruct H; apply AxiomII_P in H; apply AxiomII_P in H0.
    destruct H, H0, H1, H2; apply Theorem146 in H1.
    apply (Theorem147 _ _ z) in H1; auto; clear H H0 H2.
    unfold C in H3, H4; apply AxiomII in H3; destruct H3.
    apply AxiomII in H4; destruct H4.
    unfold Cardinal_Number in H0, H3; destruct H0, H3.
    unfold Ordinal_Number in H0, H3.
    assert (Ordinal y /\ Ordinal z).
    { unfold R in H0, H3; apply AxiomII in H0.
      apply AxiomII in H3; destruct H0, H3; split; auto. }
    apply Theorem110 in H6; destruct H6.
    + apply Theorem146 in H1; apply H5 in H0; auto; try contradiction.
    + destruct H6; auto; apply H4 in H3; auto; try contradiction.
  - apply AxiomI; split; intros; try apply Theorem19; Ens.
    apply Theorem19 in H; double H; apply Theorem140 in H0.
    destruct H0 as [f H0], H0, H1; apply AxiomII; split; auto.
    assert (WellOrdered E \{ λ x, x ≈ z /\ Ordinal x \}).
    { assert (\{ λ x, x ≈ z /\ Ordinal x \} ⊂ R).
      { unfold Included; intros; apply AxiomII in H3.
        destruct H3, H4; apply AxiomII; split; auto. }
      apply (Lemma97 _ E _) in H3; auto.
      apply Theorem107; apply Theorem113. }
    unfold WellOrdered in H3; destruct H3 as [H4 H3]; clear H4.
    assert (\{λ x, x ≈ z /\ Ordinal x\} ⊂ \{λ x, x ≈ z /\ Ordinal x\}
```

```
        /\ \{ λ x, x ≈ z /\ Ordinal x \} ≠ ∅).
    { split; try unfold Included; auto.
      apply Property_NotEmpty; exists dom(f); apply AxiomII.
      unfold Ordinal_Number, R in H2; apply AxiomII in H2; destruct H2.
      split; auto; split; auto; unfold Equivalent; exists f; auto. }
    apply H3 in H4; destruct H4; unfold FirstMember in H4; destruct H4.
    apply AxiomII in H4; destruct H4, H6.
    exists x; apply AxiomII_P.
    repeat split; try apply Theorem49; auto.
    + apply Theorem146; unfold Equivalent; Ens.
    + unfold C; apply AxiomII; split; auto.
      unfold Cardinal_Number; split; intros.
      { unfold Ordinal_Number, R; apply AxiomII; auto. }
      { unfold Less in H9; unfold R in H8.
        apply AxiomII in H8; destruct H8; intro.
        assert (y ∈ \{ λ x,x ≈ z /\ Ordinal x \}).
        { apply AxiomII; split; auto; split; auto.
          apply Theorem146 in H11; apply (Theorem147 _ x _); auto. }
        apply H5 in H12; apply H12; unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto. }
  - unfold Range; apply AxiomI; split; intros.
    + apply AxiomII in H; destruct H, H0.
      apply AxiomII_P in H0; apply H0.
    + apply AxiomII; split; Ens; exists z; apply AxiomII_P.
      repeat split; try apply Theorem49; Ens.
      apply Theorem145.
Qed.

Hint Resolve Theorem152 : set.

(* A Corollary of the Definition151 *)

Corollary Property_PClass : ∀ x, Ensemble x -> P [x] ∈ C.
Proof.
  intros.
  generalize Theorem152; intros; destruct H0, H1.
  apply Theorem19 in H; rewrite <- H1 in H.
  apply Property_Value in H; auto.
  apply Property_ran in H; rewrite H2 in H; auto.
Qed.

Hint Resolve Property_PClass : set.
```

定理 153 x 是集 $\Longrightarrow$ $P(x) \approx x$.

```
Theorem Theorem153 : ∀ x, Ensemble x -> P[x] ≈ x.
Proof.
  intros.
  generalize Theorem152; intros; destruct H0, H1.
  apply Theorem19 in H; rewrite <- H1 in H.
  apply Property_Value in H; auto.
  unfold P at 2 in H; apply AxiomII_P in H.
  apply Theorem146; apply H.
Qed.

Hint Resolve Theorem153 : set.
```

定理 154　x 和 y 均是集 $\Longrightarrow$ $(P(x) = P(y) \iff x \approx y)$.

```
Theorem Theorem154 : ∀ x y,
  Ensemble x /\ Ensemble y -> (P[x] = P[y] <-> x ≈ y).
Proof.
  intros; double H; destruct H, H0.
  apply Theorem153 in H0; apply Theorem153 in H2; split; intros.
  - rewrite H3 in H0; apply Theorem146 in H0.
    apply (Theorem147 _ P[y] _); auto.
  - generalize Theorem152; intros; destruct H4, H5.
    double H; apply Theorem19 in H; apply Theorem19 in H1.
    rewrite <- H5 in H, H1; apply Property_Value in H; auto.
    apply Property_Value in H1; auto; apply Property_ran in H1.
    rewrite H6 in H1; apply Theorem146 in H2.
    assert ([x, P [y]] ∈ P).
    { unfold P at 2; apply AxiomII_P; split.
      - apply Theorem49; split; Ens.
      - split; try apply (Theorem147 _ y _); auto. }
    unfold Function in H4; apply H4 with (x:=x); auto.
Qed.

Hint Resolve Theorem154 : set.
```

定理 155　$P(P(x)) = P(x)$.

```
Lemma Lemma155 : ∀ x, x ∈ C -> Ensemble x -> P[x] = x.
Proof.
  intros.
  double H0; apply Property_PClass in H1; AssE P[x].
  unfold C in H, H1; apply AxiomII in H; apply AxiomII in H1.
  clear H0 H2; destruct H, H1, H0, H2.
  apply Theorem153 in H; unfold Ordinal_Number in H0, H2.
  double H0; double H2; unfold R in H5, H6; apply AxiomII in H5.
```

```
  apply AxiomII in H6; destruct H5, H6; add (Ordinal P[x]) H7.
  clear H1 H5 H6 H8; apply Theorem110 in H7; destruct H7.
  + apply H4 in H1; auto; contradiction.
  + symmetry; destruct H1; auto; apply H3 in H1; auto.
    apply Theorem146 in H; contradiction.
Qed.

Theorem Theorem155 : ∀ x, P[P[x]] = P[x].
Proof.
  intros.
  generalize Theorem152; intros; destruct H, H0.
  generalize (classic (Ensemble x)); intros; destruct H2.
  - apply Property_PClass in H2; AssE P[x].
    apply Lemma155 in H2; auto.
  - generalize (classic (x ∈ dom(P))); intros; destruct H3.
    + rewrite H0 in H3; apply Theorem19 in H3; contradiction.
    + apply Theorem69 in H3; rewrite H3.
      generalize (classic (𝒰 ∈ dom(P))); intros; destruct H4.
      * generalize Theorem39; intros; elim H5; Ens.
      * apply Theorem69 in H4; rewrite H4; auto.
Qed.

Hint Resolve Lemma155 Theorem155 : set.
```

定理 156 x 是集 $\wedge\ P(x)=x \iff x\in C$.

```
Theorem Theorem156 : ∀ x,
  (Ensemble x /\ P[x] = x) <-> x∈C.
Proof.
  intros; split; intros.
  - destruct H; apply Property_PClass in H.
    rewrite H0 in H; auto.
  - AssE x; apply Lemma155 in H; auto.
Qed.

Hint Resolve Theorem156 : set.
```

定理 157 $y\in R\ \wedge\ x\subset y \implies P(x)\leqslant y$.

```
Theorem Theorem157 : ∀ x y,
  y ∈ R /\ x ⊂ y -> P[x] ≼ y.
Proof.
  intros; destruct H.
  unfold R in H; apply AxiomII in H; destruct H.
  assert (WellOrdered E x /\ WellOrdered E R).
```

```
{ split; try (apply Theorem107; apply Theorem113).
  apply Theorem107 in H1; apply (Lemma97 _ _ y); auto. }
assert (Ensemble x /\ ∼ Ensemble R).
{ split; try apply Theorem113; apply Theorem33 in H0; Ens. }
destruct H3; apply Theorem100 in H2; auto; clear H4.
destruct H2 as [f H2], H2, H4; unfold Order_PXY in H4.
destruct H4, H6, H7, H8; apply Theorem96 in H7; destruct H7.
unfold Function1_1 in H7; destruct H7 as [H11 H7]; clear H11.
generalize (Property_Relation f); intros; destruct H11.
assert (forall u, u ∈ x -> f[u] ≼ u).
{ intros; rewrite <- H5 in H13; double H13.
  apply Property_Value in H14; auto; apply Property_ran in H14.
  assert (Ordinal u /\ Ordinal f[u]).
  { rewrite H5 in H13; apply H0 in H13.
    add (u ∈ y) H1; apply Theorem111 in H1.
    unfold Section in H9; destruct H9; apply H9 in H14.
    unfold R in H14; apply AxiomII in H14; destruct H14; auto. }
  apply Theorem110 in H15; AssE ([u,f[u]]); try apply Theorem49; Ens.
  assert (Section ran(f) E R /\ Order_Pr f^-1 E E /\ On f^-1 ran(f) /\
  To f^-1 R).
  { split; auto; split; auto; split; try (split; auto).
    rewrite H12, H5; unfold Included; intros.
    apply H0 in H17; add (z ∈ y) H1; apply Theorem111 in H1.
    unfold R; apply AxiomII; split; Ens. }
  apply Theorem94 with (u:= f[u]) in H17; auto; rewrite<-H12 in H13.
  apply Lemma96''' in H13; try rewrite (Theorem61 f) in *;
  try apply H2; auto.
  rewrite <- H13 in H17; unfold LessEqual; destruct H15.
  - unfold Rrelation, E in H17; elim H17.
    apply AxiomII_P; split; auto.
  - destruct H15; try symmetry in H15; tauto. }
apply Theorem114 in H9; clear H11 H12; double H0.
try apply Theorem33 in H11; auto; apply Theorem153 in H11.
assert (x ≈ ran(f)).
{ unfold Equivalent; exists f; split; split; auto. }
assert (ran(f) ≼ y /\ Ensemble ran(f)).
{ assert (ran(f) ⊂ y).
  { unfold Included; intros.
    unfold Range in H14; apply AxiomII in H14; destruct H14, H15.
    double H15; apply Property_dom in H16; double H16.
    apply Property_Value in H17; auto; add ([x0,f[x0]] ∈ f) H15.
    unfold Function in H2; apply H2 in H15; rewrite H15 in *.
    clear H15 H17; rewrite H5 in H16; double H16; apply H13 in H16.
    unfold LessEqual in H16; destruct H16.
    - apply H0 in H15; unfold Ordinal in H1; destruct H1.
```

```
      unfold full in H17; apply H17 in H15; apply H15 in H16; auto.
     - rewrite H16; apply H0 in H15; auto. }
   split; try apply Theorem33 with (x:= y); auto.
   generalize (classic (ran(f) = y)); intros.
   unfold LessEqual; destruct H15; try tauto.
   apply Theorem108 in H14; auto; unfold Ordinal in H9; apply H9. }
  destruct H14.
  assert (WellOrdered E \{λ z, x ≈ z /\ Ordinal z\}).
  { assert (\{ λ z, x ≈ z /\ Ordinal z \} ⊂ R).
   { unfold Included; intros; apply AxiomII in H16.
     destruct H16, H17; apply AxiomII; split; auto. }
   apply (Lemma97 _ E _) in H16; auto. }
  unfold WellOrdered in H16; destruct H16 as [H17 H16]; clear H17.
  assert (\{λ z, x ≈ z /\ Ordinal z\} ⊂ \{ λ z, x ≈ z /\ Ordinal z\}
          /\ \{ λ z, x ≈ z /\ Ordinal z \} ≠ ∅).
  { split; try unfold Included; auto.
   apply Property_NotEmpty;exists ran(f);apply AxiomII;split; auto.}
  apply H16 in H17; clear H16; destruct H17; unfold FirstMember in H16.
  destruct H16; apply AxiomII in H16; destruct H16, H18.
  assert (x0 ∈ C).
  { unfold C; apply AxiomII; split; auto.
   unfold Cardinal_Number; split; intros.
   - unfold Ordinal_Number, R; apply AxiomII; Ens.
   - intro; assert (y0 ∈ \{ λ z, x ≈ z /\ Ordinal z \}).
     { unfold R in H20; apply AxiomII in H20; destruct H20.
      apply AxiomII; split; auto; split; auto.
      apply Theorem147 with (y:= x0); auto. }
     apply H17 in H23; elim H23; clear H23.
     unfold Rrelation, E; apply AxiomII_P; unfold Less in H21.
     split; auto; apply Theorem49; Ens. }
  apply Theorem156 in H20; clear H16; destruct H20.
  apply Theorem154 in H18; auto; rewrite H20 in H18; clear H20.
  assert (ran(f)∈ \{λ z, x ≈ z /\ Ordinal z\}). {apply AxiomII; Ens.}
  apply H17 in H20; clear H17; rewrite H18; unfold LessEqual.
  add (Ordinal x0) H9; apply Theorem110 in H9;
  destruct H9 as [H9|[H9 |H9]].
  - elim H20; unfold Rrelation, E; apply AxiomII_P.
   split; try apply Theorem49; Ens.
  - destruct H14.
   + unfold Ordinal in H1; destruct H1.
     unfold full in H17; apply H17 in H14; clear H17.
     unfold Included in H14; apply H14 in H9; auto.
   + rewrite H14 in H9; auto.
  - destruct H14; rewrite H9 in H14; auto.
Qed.
```

```
Hint Resolve Theorem157 : set.
```

定理 158　y 是集 $\wedge\, x \subset y \implies P(x) \leqslant P(y)$.

```
Theorem Theorem158 : ∀ x y,
  Ensemble y /\ x ⊂ y -> P[x] ≼ P[y].
Proof.
  intros; destruct H.
  assert (Ensemble x). { apply Theorem33 in H0; auto. }
  double H; apply Theorem153 in H2.
  apply Theorem146 in H2; unfold Equivalent in H2.
  destruct H2 as [f H2], H2, H3; double H.
  apply Property_PClass in H5; unfold C in H5.
  apply AxiomII in H5; destruct H5; unfold Cardinal_Number in H6.
  destruct H6; clear H7; unfold Ordinal_Number in H6.
  assert (ran(f|(x)) ⊂ P[y]).
  { rewrite <- H4; unfold Included; intros.
    unfold Range in H7; apply AxiomII in H7; destruct H7, H8.
    unfold Restriction in H8; apply Theorem4' in H8; destruct H8.
    apply Property_ran in H8; auto. }
  add (ran(f|(x)) ⊂ P [y]) H6; apply Theorem157 in H6.
  assert (x ≈ ran(f|(x))).
  { unfold Function1_1 in H2; destruct H2.
    unfold Equivalent; exists (f|(x)); split.
    - unfold Function1_1; split.
      + unfold Function; split; intros.
        * unfold Relation; intros; unfold Restriction in H9.
          apply Theorem4' in H9; destruct H9; PP H10 a b; Ens.
        * destruct H9; unfold Restriction in H9, H10.
          apply Theorem4' in H9; apply Theorem4' in H10.
          destruct H9, H10; add ([x0,z] ∈ f) H9.
          unfold Function in H2; apply H2 in H9; auto.
      + unfold Function; split; intros.
        * unfold Relation; intros; PP H9 a b; Ens.
        * destruct H9; unfold Inverse in H9, H10.
          apply AxiomII_P in H9; apply AxiomII_P in H10.
          destruct H9, H10; unfold Restriction in H11, H12.
          apply Theorem4' in H11; apply Theorem4' in H12.
          destruct H11, H12; clear H13 H14.
          assert ([x0,y0] ∈ f⁻¹ /\ [x0,z] ∈ f⁻¹).
          { unfold Inverse; split; apply AxiomII_P; Ens. }
          unfold Function in H8; apply H8 in H13; auto.
    - split; auto; apply AxiomI; intros; split; intros.
      + unfold Domain in H9; apply AxiomII in H9; destruct H9, H10.
```

```
      unfold Restriction in H10; apply Theorem4' in H10; destruct H10.
      unfold Cartesian in H11; apply AxiomII_P in H11; apply H11.
    + unfold Domain; apply AxiomII; split; Ens.
      double H9; unfold Included in H0; apply H0 in H10.
      rewrite <- H3 in H10; apply Property_Value in H10; auto.
      exists f[z]; unfold Restriction; apply Theorem4'; split; auto.
      unfold Cartesian; apply AxiomII_P; repeat split; Ens.
      AssE [z,f[z]]; apply Theorem49 in H11; destruct H11.
      apply Theorem19; auto. }
  assert (Ensemble ran(f|(x))). { apply Theorem33 in H7; auto. }
  apply Theorem154 in H8; auto; rewrite <- H8 in H6; auto.
Qed.

Hint Resolve Theorem158 : set.
```

定理 159　(x 和 y 均是集, $u \subset x$, $v \subset y$, $x \approx v$, $y \approx u$) $\Longrightarrow$ $x \approx y$.

```
Theorem Theorem159 : ∀ x y,
  Ensemble x /\ Ensemble y ->
  (∀ u v, u ⊂ x /\ v ⊂ y -> x ≈ v /\ y ≈ u -> x ≈ y).
Proof.
  intros; destruct H0, H1; elim H; intros.
  assert (Ensemble x /\ Ensemble v).
  { split; apply Theorem33 in H2; auto. }
  assert (Ensemble y /\ Ensemble u).
  { split; apply Theorem33 in H0; auto. }
  apply Theorem154 in H; apply Theorem154 in H6.
  apply Theorem154 in H7; apply H; apply H6 in H1.
  apply H7 in H3; clear H H6 H7; double H4; double H5.
  add (u ⊂ x) H4; add (v ⊂ y) H6; clear H0 H2.
  apply Theorem158 in H4; apply Theorem158 in H6.
  rewrite <- H3 in H4; rewrite <- H1 in H6; clear H1 H3.
  apply Property_PClass in H; apply Property_PClass in H5.
  unfold C in H, H5; apply AxiomII in H; apply AxiomII in H5.
  destruct H, H5; unfold Cardinal_Number in H0, H2.
  destruct H0, H2; unfold Ordinal_Number, R in H0, H2.
  apply AxiomII in H0; apply AxiomII in H2; destruct H0, H2.
  clear H H0 H1 H2 H3 H5.
  assert (Ordinal P [x] /\ Ordinal P [y]). { auto. }
  assert (Ordinal P [y] /\ Ordinal P [x]). { auto. }
  apply Theorem118 in H; apply Theorem118 in H0; clear H7 H8.
  apply H in H6; apply H0 in H4; clear H H0.
  apply Theorem27; split; auto.
Qed.
```

```
Hint Resolve Theorem159 : set.
```

上面的结论就是著名的 Cantor-Bernstein-Schroeder 定理, 这里给出的是依赖选择公理的机器证明过程. 下面给出一个不依赖选择公理而更直接的证明 [41, 84, 94].

```
Definition Imgset f x := \{ λ u, ∃ v, v ∈ x /\ u = f[v] \}.

Inductive Ind := | fir : Ind | next : Ind -> Ind.

Fixpoint C' x u g f (n: Ind) : Class :=
   match n with
      | fir => (x ~ u)
      | next p => Imgset g (Imgset f (C' x u g f p))
       end.

Lemma Lemma_CBS1 : ∀ a b f, Function1_1 f ->
  b ∈ ran(f) -> a = f⁻¹[b] -> b = f [a].
Proof.
  intros. pattern b.
  rewrite Lemma96''' with (f:=f); try apply H; auto.
  rewrite H1; auto.
Qed.

Lemma Lemma_CBS2 : ∀ a b f, Function1_1 f ->
  a ∈ dom(f) -> b = f [a] -> a = f⁻¹[b].
Proof.
  intros. pattern a.
  rewrite Lemma_CBS1 with (f:=f⁻¹) (a:=b); auto.
  - destruct H; split; auto.
    rewrite Theorem61; try apply H; auto.
  - rewrite <- Lemma96; auto.
  - rewrite Theorem61; try apply H; auto.
Qed.

Theorem Cantor_Bernstein_Schroeder : ∀ x y u v,
  Ensemble x -> Ensemble y -> u ⊂ x -> v ⊂ y ->
  x ≈ v -> y ≈ u -> x ≈ y.
Proof.
  intros; destruct H3 as [f H3], H3, H5, H4 as [g H4], H4, H7.
  set (C:= (C' x u g f)); set (CC:= \{ λ u, exists n, u = C n \}).
  assert (forall z, z ∈ x -> ~ z ∈ (⋃ CC) -> z ∈ (x~(⋃ CC))) as G1;
  intros.
  { apply AxiomII; repeat split; Ens.
    apply AxiomII; split; Ens. }
  assert ((⋃ CC) ⊂ x) as G2.
```

```
{ red; intros.
  apply AxiomII in H9; destruct H9, H10, H10.
  apply AxiomII in H11; destruct H11, H12; subst x0.
  clear H9 H11; generalize dependent z.
  induction x1; intros; unfold C in H10; simpl in H10.
  - apply AxiomII in H10; tauto.
  - apply AxiomII in H10; destruct H10, H10, H10.
    apply AxiomII in H10; destruct H10, H12, H12.
    apply IHx1 in H12; subst x0 x v y u z.
    apply Property_Value in H12; try apply H3.
    apply Property_ran in H12; apply H2 in H12.
    apply Property_Value in H12; try apply H4.
    apply Property_ran in H12; auto. }
assert (forall z, z ∈ x -> ~ z ∈ (⋃ CC) -> z ∈ ran( g)) as G3;
intros.
{ rewrite H8; destruct (classic (z ∈ u)); auto.
  elim H10; apply AxiomII; split; Ens.
  exists (C fir); unfold C; simpl; split.
  - apply AxiomII; repeat split; Ens.
    apply AxiomII; split; Ens.
  - apply AxiomII; split.
    + apply Theorem33 with (x:=x); auto.
      red; intros; apply AxiomII in H12; tauto.
    + exists fir; auto. }
exists \{\ λ p q, (p ∈ x) /\
  ((p ∈ (⋃CC) -> q = f[p]) /\ (p ∈ (x ~ (⋃CC)) -> q=g⁻¹[p]))\}\.
repeat split; intros.
- red; intros; PP H9 a b; eauto.
- destruct H9; apply AxiomII_P in H9; destruct H9, H11, H12.
  apply AxiomII_P in H10;
  destruct H10, H14, H15, (classic (x0 ∈ (⋃ CC))).
  + rewrite H12, H15; auto.
  + rewrite H13, H16; auto.
- red; intros; PP H9 a b; eauto.
- destruct H9; apply AxiomII_P in H9; destruct H9.
  apply AxiomII_P in H10; destruct H10.
  apply AxiomII_P in H11; apply AxiomII_P in H12.
  destruct H11, H13, H14, H12, H16, H17.
  destruct (classic (y0 ∈ (⋃ CC))), (classic (z ∈ (⋃ CC))).
  + apply H14 in H19; apply H17 in H20; subst x.
    apply Lemma_CBS2 in H19; apply Lemma_CBS2 in H20; auto.
    rewrite H19, H20; auto.
  + double H19; double H20; apply G1 in H20; auto.
    apply H14 in H19; apply H18 in H20.
    subst x; rewrite H19 in H20.
```

```
  apply G3 in H16; apply Lemma_CBS1 in H20; auto.
  elim H22; apply AxiomII; split; Ens.
  apply AxiomII in H21; destruct H21, H21, H21.
  apply AxiomII in H23; destruct H23, H24; subst x.
  exists (C (next x1)); split.
  * unfold C; simpl; apply AxiomII; split; Ens.
    exists f[y0]; split; auto; apply AxiomII; split; Ens.
    apply Theorem69 in H13; apply Theorem19; auto.
  * apply AxiomII; split; Ens; unfold C; simpl.
    apply Theorem33 with (x:= dom(f)); auto; red; intros.
    apply AxiomII in H24; destruct H24, H25, H25; subst z0.
    assert (x ∈ dom(g)).
    { destruct (classic (x ∈ dom(g))); auto.
      apply Theorem69 in H26; rewrite H26 in H24.
      destruct (Theorem39 H24). }
    apply Property_Value in H26; try apply H4.
    apply Property_ran in H26.
    rewrite H8 in H26; auto.
+ double H19; double H20; apply G1 in H19; auto.
  apply H17 in H20; apply H15 in H19.
  subst x; rewrite H20 in H19.
  apply G3 in H13; apply Lemma_CBS1 in H19; auto.
  elim H21; apply AxiomII; split; Ens.
  apply AxiomII in H22; destruct H22, H22, H22.
  apply AxiomII in H23; destruct H23, H24; subst x.
  exists (C (next x1)); split.
  * unfold C; simpl; apply AxiomII; split; Ens.
    exists f[z]; split; auto; apply AxiomII; split; Ens.
    apply Theorem69 in H16; apply Theorem19; auto.
  * apply AxiomII; split; Ens; unfold C; simpl.
    apply Theorem33 with (x:= dom(f)); auto; red; intros.
    apply AxiomII in H24; destruct H24, H25, H25; subst z0.
    assert (x ∈ dom(g)).
    { destruct (classic (x ∈ dom(g))); auto.
      apply Theorem69 in H26; rewrite H26 in H24.
      destruct (Theorem39 H24). }
    apply Property_Value in H26; try apply H4.
    apply Property_ran in H26.
    rewrite H8 in H26; auto.
+ double H13; double H16.
  apply G3 in H21; apply G3 in H22; auto.
  apply G1 in H19; apply G1 in H20; auto.
  apply H15 in H19; apply H18 in H20; auto.
  apply Lemma_CBS1 in H19; apply Lemma_CBS1 in H20; auto.
  rewrite H19, H20; auto.
```

```
- apply AxiomI; split; intros.
  + apply AxiomII in H9; destruct H9, H10.
    apply AxiomII_P in H10; destruct H10; tauto.
  + apply AxiomII; split; Ens.
    destruct (classic (z ∈ (⋃ CC))); subst x.
    * exists f[z]; apply AxiomII_P.
      repeat split; intros; auto.
      { apply Theorem49; split; Ens.
        apply Theorem19; apply Theorem69; auto. }
      { apply AxiomII in H5; destruct H5, H11.
        apply AxiomII in H12; destruct H12; contradiction. }
    * exists g ⁻¹[z]; apply AxiomII_P.
      repeat split; intros; auto; try tauto.
      apply Theorem49; split; Ens; apply Theorem19; apply Theorem69.
      rewrite <- Lemma96', H8.
      destruct (classic (z ∈ u)); auto.
      elim H10; apply AxiomII; split; Ens.
      exists (C fir); split; apply AxiomII; repeat split; Ens.
      -- apply AxiomII; split; Ens.
      -- apply Theorem33 with dom(f); auto.
         red; intros; apply AxiomII in H11; tauto.
- apply AxiomI; split; intros.
  + apply AxiomII in H9; destruct H9, H10.
    apply AxiomII_P in H10; destruct H10, H11, H12.
    destruct (classic (x0 ∈ (⋃ CC))).
    * apply H12 in H14; subst z.
      apply H2; rewrite <- H6; rewrite <- H5 in H11.
      apply Property_Value in H11; try apply H3.
      apply Property_ran in H11; auto.
    * double H11; apply G1 in H11; apply G3 in H15; auto.
      apply H13 in H11; auto; rewrite Lemma96' in H15.
      apply Property_Value in H15; try apply H4.
      apply Property_ran in H15; subst z.
      rewrite <- Lemma96, H7 in H15; auto.
  + apply AxiomII; split; Ens.
    destruct (classic (z ∈ (Imgset f (⋃ CC)))).
    * apply AxiomII in H10; destruct H10, H11, H11.
      exists x0; apply AxiomII_P; repeat split; auto; intros.
      { apply Theorem49; split; Ens. }
      { apply AxiomII in H13; destruct H13, H14.
        apply AxiomII in H15; destruct H15; contradiction. }
    * assert (g[z] ∈ ran(g)).
      { subst y; apply Property_Value in H9; try apply H4.
        apply Property_ran in H9; Ens. }
      assert (forall n, (C n) ⊂ (⋃ CC)); intros.
```

```
{ apply Theorem32; apply AxiomII; split; Ens.
  apply Theorem33 with (x:=x); auto.
  red; induction n; intros.
  - apply AxiomII in H12; tauto.
  - unfold C in H12; simpl in H12.
    apply AxiomII in H12; destruct H12, H13, H13.
    apply AxiomII in H13; destruct H13, H15, H15.
    apply IHn in H15; subst x z0 x0 v y u.
    apply Property_Value in H15; try apply H3.
    apply Property_ran in H15; apply H2 in H15.
    apply Property_Value in H15; try apply H4.
    apply Property_ran in H15; auto. }
assert (~ g[z] ∈ (⋃ CC)); try intro.
{ apply AxiomII in H13; destruct H13, H14, H14.
  apply AxiomII in H15; destruct H15, H16; subst x0 u.
  destruct x1.
  - apply AxiomII in H14; destruct H14, H14.
    apply AxiomII in H16; apply H16; auto.
  - unfold C in H14; simpl in H14.
    apply AxiomII in H14; destruct H14, H14, H14.
    apply AxiomII in H14; destruct H14, H17, H17.
    assert (z = x0) as G4.
    { rewrite Lemma96''' with (f:=g⁻¹); try apply H4.
      - pattern z; rewrite Lemma96''' with (f:=g⁻¹);
        try apply H4.
        + repeat rewrite Theorem61; try apply H4;
          rewrite H16; auto.
        + rewrite Theorem61; apply H4.
        + rewrite <- Lemma96, H7; auto.
      - rewrite Theorem61; apply H4.
      - rewrite <- Lemma96; subst x x0 y v.
        apply H12 in H17; apply G2 in H17.
        apply Property_Value in H17; try apply H3.
        apply Property_ran in H17; auto. }
    subst x0 x; apply H10; apply AxiomII; split; Ens.
    exists x2; split; auto.
    apply AxiomII; split; Ens.
    exists (C x1); split; auto.
    apply AxiomII; split; eauto.
    apply Theorem33 with (x:=⋃ CC); auto.
    apply Theorem33 with (x:=dom(f)); auto. }
exists g[z]; apply AxiomII_P; repeat split; intros.
{ apply Theorem49; split; Ens. }
{ subst u x; auto. }
{ contradiction. }
```

```
    { pattern g at 2; rewrite <- Theorem61; try apply H4.
      rewrite <- Lemma96'''; auto; try apply H4.
      - rewrite Theorem61; apply H4.
      - apply Property_Value' in H11; try apply H4.
        apply Property_dom in H11; rewrite <- Lemma96; auto. }
Qed.

Hint Resolve Cantor_Bernstein_Schroeder : set.
```

定理 160 f 是函数 $\Longrightarrow P(f$ 的值域 $)\leqslant P(f$ 的定义域 $)$.

```
Definition En_g f c : Class :=
  \{\ λ v u, v ∈ ran(f) /\ u = c[\{ λ x, v = f[x] \}] \}\.

Lemma Lemma160 : ∀ c f y,
  Function f -> Ensemble dom(f) -> ChoiceFunction c -> dom( c) = 𝒰 ~
  [∅] -> y ∈ ran( f) -> \{ λ x, y = f[x] \} ∈ dom(c).
Proof.
  intros.
  unfold ChoiceFunction in H1; destruct H1.
  unfold Range in H3; apply AxiomII in H3; destruct H3, H5.
  rewrite H2; unfold Setminus; apply Theorem4'.
  assert (Ensemble (\{ λ x, y = f[x] \})).
  { apply Theorem33 with (x:= dom(f)); auto.
    unfold Included; intros; apply AxiomII in H6; destruct H6.
    rewrite H7 in H5; clear H7; apply Property_ran in H5.
    apply Property_Value' in H5; auto; apply Property_dom in H5; auto.}
  split; try apply Theorem19; auto.
  unfold Complement; apply AxiomII; split; auto.
  unfold NotIn; intro; unfold Singleton in H7.
  apply AxiomII in H7; clear H6; destruct H7.
  assert (∅ ∈ 𝒰).
  { apply Theorem19; generalize AxiomVIII; intros.
    destruct H8; Ens; exists x0; apply H8. }
  apply H7 in H8; clear H7.
  assert (x ∈ \{ λ x, y = f [x] \}).
  { apply AxiomII; double H5; apply Property_dom in H7.
    split; Ens; apply Property_Value in H7; auto.
    unfold Function in H; apply H with (x:=x); Ens. }
  rewrite H8 in H7; generalize (Theorem16 x); intros; contradiction.
Qed.

Theorem Theorem160 : ∀ f,
  Function f -> P[ran(f)] ≼ P[dom(f)].
Proof.
```

```
intros.
generalize (classic (Ensemble dom(f))); intros; destruct H0.
- generalize AxiomIX; intros; destruct H1 as [c H1], H1.
  assert (Function1_1 (En_g f c)).
  { unfold Function1_1, Function; repeat split; intros.
    - unfold Relation; intros; PP H3 a b; Ens.
    - unfold En_g in H3; destruct H3.
      apply AxiomII_P in H3; apply AxiomII_P in H4.
      destruct H3, H4, H5, H6; rewrite H7, H8; auto.
    - unfold Relation; intros; PP H3 a b; Ens.
    - unfold Inverse, En_g in H3; destruct H3.
      apply AxiomII_P in H3; apply AxiomII_P in H4.
      destruct H3, H4; clear H3 H4; apply AxiomII_P in H5.
      apply AxiomII_P in H6; destruct H5, H6, H4, H6.
      assert (\{λ x, y=f[x]\} ∈ dom(c) /\ \{λ x, z=f[x]\} ∈ dom(c)).
      { split; apply Lemma160; auto. }
      destruct H9; apply H1 in H9; apply H1 in H10.
      rewrite <- H7 in H9; rewrite <- H8 in H10; apply AxiomII in H9.
      apply AxiomII in H10;destruct H9, H10;rewrite H11, H12;auto.}
  assert (ran(En_g f c) ⊂ dom(f)).
  { unfold Included; intros; unfold Range, En_g in H4; unfold Domain.
    apply AxiomII in H4; destruct H4, H5; apply AxiomII_P in H5.
    destruct H5, H6; apply AxiomII; split; auto; exists f[z].
    assert (\{ λ x0, x=f [x0] \}∈dom(c)). {apply Lemma160; auto.}
    apply H1 in H8; rewrite <- H7 in H8;
    apply AxiomII in H8; destruct H8.
    rewrite H9 in H6; clear H9; apply Property_Value'; auto. }
  assert (Ensemble dom(f) /\ ran(En_g f c) ⊂ dom(f)); auto.
  apply Theorem158 in H5; auto.
  assert (dom(En_g f c) ≈ ran(En_g f c)).
  { unfold Equivalent; exists (En_g f c); auto. }
  assert (dom(En_g f c) = ran(f)).
  { apply AxiomI; split; intros.
    - unfold Domain, En_g in H7; apply AxiomII in H7.
      destruct H7, H8; apply AxiomII_P in H8; apply H8.
    - unfold Domain, En_g; apply AxiomII; split; Ens.
      exists c[\{ λ x, z = f [x] \}]; apply AxiomII_P; split; auto.
      apply Theorem49; split; Ens; exists \{ λ x, z = f [x] \}.
      unfold ChoiceFunction in H1; apply H1; apply Lemma160; auto. }
  rewrite H7 in H6; clear H7.
  assert (Ensemble ran(f) /\ Ensemble ran(En_g f c)).
  { double H0; split; try apply AxiomV in H0; auto.
    apply Theorem33 with (x:= dom(f)); auto. }
  apply Theorem154 in H7; apply H7 in H6; clear H7; rewrite H6; auto.
- generalize Theorem152; intros; destruct H1, H2.
```

```
    generalize (classic (dom(f) ∈ dom(P))); intros; destruct H4.
    + rewrite H2 in H4; apply Theorem19 in H4; contradiction.
    + apply Theorem69 in H4; rewrite H4; clear H4.
      generalize (classic (ran(f) ∈ dom(P))); intros; destruct H4.
      * rewrite H2 in H4; apply Theorem19 in H4.
        apply Property_PClass in H4; unfold LessEqual.
        left; apply Theorem19; Ens.
      * apply Theorem69 in H4; rewrite H4; unfold LessEqual; tauto.
Qed.

Hint Resolve Theorem160 : set.
```

这里的定理 160 较文献 [41] 中的对应定理条件要广泛一些, 原定理中的 “f 是集” 这一条件可以去掉.

定理 161 x 是集 $\Longrightarrow$ $P(x) < P(2^x)$.

```
Theorem Theorem161 : ∀ x,
  Ensemble x -> P[x] ≺ P[ pow(x) ].
Proof.
  intros.
  assert (x ≈ \{ λ v, ∃ u, u ∈ x /\ v = [u] \}).
  { unfold Equivalent; exists \{\ λ u v, u ∈ x /\ v = [u] \}\.
    repeat split; auto; unfold Relation; intros; try PP H0 a b; Ens.
    - destruct H0; apply AxiomII_P in H0; apply AxiomII_P in H1.
      destruct H0, H1, H2, H3; rewrite H4, H5; auto.
    - destruct H0; apply AxiomII_P in H0; apply AxiomII_P in H1.
      destruct H0, H1; apply AxiomII_P in H2; apply AxiomII_P in H3.
      clear H0 H1; destruct H2, H3, H1, H3; rewrite H4 in H5.
      assert (y ∈ [y]). { apply AxiomII; split; Ens. }
      rewrite H5 in H6; apply AxiomII in H6; destruct H6.
      apply H7; apply Theorem19; Ens.
    - apply AxiomI; split; intros.
      + unfold Domain in H0; apply AxiomII in H0; destruct H0, H1.
        apply AxiomII_P in H1; apply H1.
      + unfold Domain; apply AxiomII; split; Ens; exists [z].
        AssE z; apply Theorem42 in H1; apply AxiomII_P.
        repeat split; try apply Theorem49; Ens.
    - apply AxiomI; split; intros.
      + unfold Range in H0; apply AxiomII in H0; destruct H0, H1.
        apply AxiomII_P in H1; destruct H1, H2; apply AxiomII; Ens.
      + apply AxiomII in H0; destruct H0, H1, H1; unfold Range.
        apply AxiomII; split; auto; exists x0; apply AxiomII_P.
        repeat split; try apply Theorem49; Ens. }
  assert (Ensemble pow(x) /\ \{λ v, ∃ u, u ∈ x /\ v=[u]\} ⊂ pow(x)).
  { split; try apply Theorem38 in H; auto.
```

```
    unfold Included; intros; apply AxiomII in H1; destruct H1, H2, H2.
    rewrite H3 in *; clear H3; unfold PowerSet;
    apply AxiomII; split; auto.
    unfold Included; intros; apply AxiomII in H3; destruct H3.
    rewrite H4; try apply Theorem19; Ens. }
  assert (Ensemble x /\ Ensemble \{λ v, ∃ u, u ∈ x /\ v=[u]\}).
  { split; auto; destruct H1; apply Theorem33 in H2; auto. }
  apply Theorem158 in H1; apply Theorem154 in H2; apply H2 in H0.
  rewrite <- H0 in H1; clear H0 H2; unfold LessEqual in H1.
  unfold Less; destruct H1; auto.
  assert (Ensemble x /\ Ensemble pow(x)).
  { split; auto; apply Theorem38 in H; auto. }
  apply Theorem154 in H1; apply H1 in H0; clear H1.
  unfold Equivalent in H0; destruct H0 as [f H0], H0, H1.
  assert (\{λ v, v ∈ x /\ v ∉ f[v]\} ∈ ran(f)).
  { assert (\{λ v, v ∈ x /\ v ∉ f[v]\} ⊂ x).
    { unfold Included; intros; apply AxiomII in H3; apply H3. }
    double H3; apply Theorem33 in H4; auto; rewrite H2.
    unfold PowerSet; apply AxiomII; split; auto. }
  unfold Range in H3; apply AxiomII in H3; destruct H3, H4 as [u H4].
  double H4; apply Property_dom in H5; unfold Function1_1 in H0.
  destruct H0; clear H6; double H5; apply Property_Value in H6; auto.
  rewrite H1 in H5; add ([u,f[u]] ∈ f) H4; apply H0 in H4; clear H6.
  generalize (classic (u ∈ f[u])); intros; destruct H6.
  - double H6; rewrite <- H4 in H7; apply AxiomII in H7.
    destruct H7, H8; contradiction.
  - elim H6; rewrite <- H4; apply AxiomII; Ens.
Qed.

Hint Resolve Theorem161 : set.
```

定理 162 C 不是集.

```
Theorem Theorem162 : ~ Ensemble C.
Proof.
  intro.
  apply AxiomVI in H; double H.
  apply Theorem38 in H0; try apply C.
  apply Property_PClass in H0.
  assert (Ensemble ( ⋃ C) /\ P[ pow( ⋃ C ) ] ⊂ ⋃ C).
  { split; auto; apply Theorem32 in H0; apply H0. }
  apply Theorem158 in H1; rewrite Theorem155 in H1.
  double H; apply Theorem161 in H2; unfold Less in H2.
  unfold LessEqual in H1; destruct H1.
  - apply Property_PClass in H; generalize Theorem150; intros.
```

```
    apply Theorem88 in H3; destruct H3; unfold Asymmetric in H4.
    assert (P[ ⋃ C] ∈ C /\ P[pow( ⋃ C)] ∈ C /\
      Rrelation P[ ⋃ C] E P [pow( ⋃ C)]).
    { repeat split; auto; unfold Rrelation, E.
      apply AxiomII_P; split; try apply Theorem49; Ens. }
    apply H4 in H5; apply H5; clear H4 H5.
    unfold Rrelation, E; apply AxiomII_P.
    split; try apply Theorem49; Ens.
  - rewrite H1 in H2.
    generalize (Theorem101 P[ ⋃ C]); intros; contradiction.
Qed.

Hint Resolve Theorem162 : set.
```

上面定理的证明就其结构而言类似于 Russell 悖论.

定理 163 $(x \in \omega,\ y \in \omega,\ x+1 \approx y+1) \implies x \approx y.$

```
Ltac SplitEns := apply AxiomII; split; Ens.

Ltac SplitEnsP := apply AxiomII_P; split; try apply Theorem49; Ens.

Definition En_g' f x y : Class :=
  \{\ λ u v, [u,v] ∈ (f ∼ ([[x,f[x]]] ∪ [[f⁻¹[y],y]])) \/
  [u,v] = [f⁻¹[y],f[x]] \/ [u,v] = [x,y] \}\.

Theorem Theorem163 : forall x y,
  x ∈ ω -> y ∈ ω -> (PlusOne x) ≈ (PlusOne y) -> x ≈ y.
Proof.
  intros.
  unfold Equivalent in H1; destruct H1 as [f H1], H1, H2.
  unfold Function1_1 in H1; destruct H1; unfold Equivalent.
  exists ((En_g' f x y) | (x)); repeat split; intros.
  - unfold Relation; intros; unfold Restriction in H5.
    apply Theorem4' in H5; destruct H5; PP H6 a b; Ens.
  - destruct H5; unfold Restriction in H5, H6.
    apply Theorem4' in H5; apply Theorem4' in H6.
    destruct H5, H6; clear H8; unfold En_g' in H5, H6.
    apply AxiomII_P in H5; apply AxiomII_P in H6; destruct H5,H6.
    unfold Cartesian in H7; apply AxiomII_P in H7; clear H5.
    destruct H7, H7; clear H10; destruct H8, H9.
    + unfold Setminus in H8, H9; apply Theorem4' in H8.
      apply Theorem4' in H9; destruct H8, H9; clear H10 H11.
      unfold Function in H1; apply H1 with (x:= x0); auto.
    + destruct H9.
```

```
    * unfold Setminus in H8; apply Theorem4' in H8; destruct H8.
      unfold Complement in H10; apply AxiomII in H10; clear H5.
      destruct H10; elim H10; clear H10; apply Theorem4.
      right; apply AxiomII; split; auto; intros; clear H10.
      apply Theorem49 in H6; apply Theorem55 in H9; auto.
      destruct H9; clear H10; double H8; apply Property_dom in H10.
      apply Property_Value in H10; auto; add ([x0,f[x0]] ∈ f) H8.
      apply H1 in H8; clear H10; rewrite H9 in H8.
      rewrite <- Lemma96''' in H8; auto. rewrite H8, H9; auto.
      rewrite H3; unfold PlusOne; apply Theorem4; right.
      unfold Singleton; apply AxiomII; split; Ens.
    * apply Theorem49 in H6;apply Theorem55 in H9;auto;destruct H9.
      rewrite H9 in H7; generalize (Theorem101 x); contradiction.
  + destruct H8.
    * unfold Setminus in H9; apply Theorem4' in H9; destruct H9.
      unfold Complement in H10; apply AxiomII in H10; clear H6.
      destruct H10; elim H10; clear H10; apply Theorem4.
      right; apply AxiomII; split; auto; intros; clear H10.
      apply Theorem49 in H5; apply Theorem55 in H8; auto.
      destruct H8; clear H10; double H9; apply Property_dom in H10.
      apply Property_Value in H10; auto; add ([x0,f[x0]] ∈ f) H9.
      apply H1 in H9; clear H10; rewrite H8 in H9.
      rewrite <- Lemma96''' in H9; auto. rewrite H8, H9; auto.
      rewrite H3; unfold PlusOne; apply Theorem4; right.
      unfold Singleton; apply AxiomII; split; Ens.
    * apply Theorem49 in H5;apply Theorem55 in H8;auto;destruct H8.
      rewrite H8 in H7; generalize (Theorem101 x); contradiction.
  + apply Theorem49 in H5; apply Theorem49 in H6.
    destruct H8, H9; apply Theorem55 in H8;
    apply Theorem55 in H9; auto.
    * destruct H8, H9; rewrite H10, H11; auto.
    * destruct H9; rewrite H9 in H7.
      generalize (Theorem101 x); intros; contradiction.
    * destruct H8; rewrite H8 in H7.
      generalize (Theorem101 x); intros; contradiction.
    * destruct H8; rewrite H8 in H7.
      generalize (Theorem101 x); intros; contradiction.
- unfold Relation; intros; PP H5 a b; Ens.
- destruct H5; unfold Inverse, Restriction in H5, H6.
  apply AxiomII_P in H5; apply AxiomII_P in H6; destruct H5, H6.
  apply Theorem4' in H7; apply Theorem4' in H8; destruct H7, H8.
  apply AxiomII_P in H7; apply AxiomII_P in H8; destruct H7, H8.
  unfold Cartesian in H9; apply AxiomII_P in H9; clear H7.
  destruct H9, H9; clear H13; apply AxiomII_P in H10; clear H8.
  destruct H10, H10; clear H13; destruct H11, H12.
```

```
+ unfold Setminus in H11, H12; apply Theorem4' in H11.
  apply Theorem4' in H12; destruct H11, H12; clear H13 H14.
  assert ([x0,y0] ∈ f⁻¹ /\ [x0,z] ∈ f⁻¹).
  { unfold Inverse; split; apply AxiomII_P; split; auto. }
  unfold Function in H4; apply H4 in H13; auto.
+ destruct H12.
  * unfold Setminus in H11; apply Theorem4' in H11; destruct H11.
    clear H13; apply Theorem49 in H8; apply Theorem55 in H12; auto.
    destruct H12; rewrite H13 in *; double H11.
    apply Property_ran in H14; apply Property_Value' in H14; auto.
    assert ([f[x],y0] ∈ f⁻¹ /\ [f[x],x] ∈ f⁻¹).
    { unfold Inverse; split; apply AxiomII_P;
      split; auto; AssE [x,f [x]].
      apply Theorem49 in H15; destruct H15; apply Theorem49; auto.}
    unfold Function in H4; apply H4 in H15; auto.
    rewrite H15 in H9; generalize (Theorem101 x); contradiction.
  * unfold Setminus in H11; apply Theorem4' in H11; destruct H11.
    unfold Complement in H13; apply AxiomII in H13; clear H7.
    destruct H13; elim H13; clear H13; apply Theorem4.
    right; apply AxiomII; split; auto; intros; clear H13.
    apply Theorem49 in H8; apply Theorem55 in H12; auto.
    destruct H12; rewrite H13 in *; clear H6 H8 H12 H13.
    assert ([y,y0] ∈ f⁻¹). { apply AxiomII_P; Ens. }
    double H6; apply Property_dom in H8;
    apply Property_Value in H8; auto.
    add ([y,y0] ∈ f⁻¹) H8; apply H4 in H8; rewrite H8; auto.
+ destruct H11.
  * unfold Setminus in H12; apply Theorem4' in H12; destruct H12.
    clear H13; apply Theorem49 in H7; apply Theorem55 in H11; auto.
    destruct H11; rewrite H13 in *; double H12.
    apply Property_ran in H14; apply Property_Value' in H14; auto.
    assert ([f[x],z] ∈ f⁻¹ /\ [f[x],x] ∈ f⁻¹).
    { unfold Inverse; split;
      apply AxiomII_P; split; auto; AssE [x, f[x]].
      apply Theorem49 in H15; destruct H15; apply Theorem49; auto.}
    unfold Function in H4; apply H4 in H15; auto.
    rewrite H15 in H10; generalize (Theorem101 x); contradiction.
  * unfold Setminus in H12; apply Theorem4' in H12; destruct H12.
    unfold Complement in H13; apply AxiomII in H13; clear H8.
    destruct H13; elim H13; clear H13; apply Theorem4.
    right; apply AxiomII; split; auto; intros; clear H13.
    apply Theorem49 in H7; apply Theorem55 in H11; auto.
    destruct H11; rewrite H13 in *; clear H5 H7 H11 H13.
    assert ([y,z] ∈ f⁻¹). { apply AxiomII_P; Ens. }
    double H5; apply Property_dom in H7;
```

```
    apply Property_Value in H7; auto.
    add ([y,z] ∈ f⁻¹) H7; apply H4 in H7; rewrite H7; auto.
  + apply Theorem49 in H7; apply Theorem49 in H8.
    destruct H11, H12; apply Theorem55 in H11;
    apply Theorem55 in H12; auto.
    * destruct H11, H12; rewrite H11, H12; auto.
    * destruct H12; rewrite H12 in H10.
      generalize (Theorem101 x); intros; contradiction.
    * destruct H11; rewrite H11 in H9.
      generalize (Theorem101 x); intros; contradiction.
    * destruct H12; rewrite H12 in H10.
      generalize (Theorem101 x); intros; contradiction.
- apply AxiomI; split; intros.
  + unfold Domain in H5; apply AxiomII in H5; destruct H5, H6.
    unfold Restriction in H6; apply Theorem4' in H6; destruct H6.
    unfold Cartesian in H7; apply AxiomII_P in H7; apply H7.
  + unfold Domain; apply AxiomII; split; Ens.
    assert ([x,f[x]] ∈ f).
    { apply Property_Value; auto; rewrite H2; unfold PlusOne.
      apply Theorem4; right; apply AxiomII; split; Ens. }
    generalize (classic (z = f⁻¹[y])); intros; destruct H7.
    * rewrite H7 in *; AssE [x,f[x]]; clear H6 H7.
      apply Theorem49 in H8; destruct H8.
      exists f[x]; unfold Restriction; apply Theorem4'.
      split; SplitEnsP; split; try apply Theorem19; auto.
    * assert (z ∈ dom(f)). { rewrite H2; apply Theorem4; tauto.}
      apply Property_Value in H8; auto; AssE [z,f[z]].
      apply Theorem49 in H9; destruct H9; exists f[z].
      unfold Restriction; apply Theorem4'; split; SplitEnsP.
      { left; unfold Setminus; apply Theorem4'; split; auto.
        unfold Complement; apply AxiomII; split; Ens.
        intro; apply Theorem4 in H11; destruct H11.
        - apply AxiomII in H11; destruct H11; clear H11.
          assert ([x,f[x]] ∈ 𝒰). { apply Theorem19; Ens. }
          apply H12 in H11; clear H12; apply Theorem55 in H11; auto.
          destruct H11; rewrite H11 in H5;
          generalize (Theorem101 x); auto.
        - apply AxiomII in H11; destruct H11; clear H11.
          assert ([(f⁻¹)[y],y] ∈ 𝒰).
          { apply Theorem19; Ens; exists f.
            assert (y ∈ ran(f)).
            { rewrite H3; unfold PlusOne; apply Theorem4; right.
              apply AxiomII; split; Ens. }
            rewrite Lemma96' in H11;
            apply Property_Value in H11; auto.
```

```
        apply AxiomII_P in H11; apply H11. }
       apply H12 in H11; clear H12; apply Theorem55 in H11; auto.
       destruct H11; contradiction. }
     { split; try apply Theorem19; auto. }
 - apply AxiomI; split; intros.
   + unfold Range in H5; apply AxiomII in H5; destruct H5, H6.
     unfold Restriction in H6; apply Theorem4' in H6; destruct H6.
     unfold Cartesian in H7; apply AxiomII_P in H7; destruct H7.
     clear H7; destruct H8; clear H8; unfold En_g' in H6.
     apply AxiomII_P in H6; destruct H6, H8 as [H8|[H8|H8]].
     * unfold Setminus in H8; apply Theorem4' in H8; destruct H8.
       unfold Complement in H9; apply AxiomII in H9; clear H6.
       destruct H9; double H8; apply Property_ran in H10;
       rewrite H3 in H10.
       unfold PlusOne in H10;apply Theorem4 in H10;destruct H10;auto.
       apply AxiomII in H10; clear H5; destruct H10.
       rewrite H10 in *; try apply Theorem19; Ens; clear H10.
       double H8; apply Property_ran in H10; rewrite Lemma96' in H10.
       apply Property_Value in H10; auto; apply Theorem49 in H6.
       destruct H6; clear H11; add ([y,x0] ∈ f⁻¹) H10; try SplitEnsP.
       apply H4 in H10; rewrite H10 in H9; elim H9.
       apply Theorem4; right; SplitEns.
     * apply Theorem49 in H6; apply Theorem55 in H8; auto;destruct H8.
       assert (x ∈ dom(f)).
       { rewrite H2; unfold PlusOne; apply Theorem4; right.
         unfold Singleton; apply AxiomII; split; Ens. }
       double H10; apply Property_Value in H11; auto.
       apply Property_ran in H11; rewrite H3 in H11;
       unfold PlusOne in H11.
       apply Theorem4 in H11; rewrite H9 in *; destruct H11; auto.
       apply AxiomII in H11; clear H5; destruct H11.
       rewrite <- H11 in H8; try apply Theorem19; Ens.
       pattern f at 2 in H8; rewrite <- Theorem61 in H8; try apply H1.
       rewrite <- Lemma96''' in H8; try rewrite Theorem61;
       try apply H1; auto.
       { rewrite H8 in H7; generalize (Theorem101 x); contradiction. }
       { rewrite <- Lemma96; auto. }
     * apply Theorem49 in H6;apply Theorem55 in H8;auto;destruct H8.
       rewrite H8 in H7; generalize (Theorem101 x); contradiction.
   + unfold Range; apply AxiomII; split; Ens.
     assert (z∈ran(f)).
     { rewrite H3; unfold PlusOne; apply Theorem4; auto. }
     generalize (classic (z = f[x])); intros; destruct H7.
     * rewrite H7 in *; clear H7.
       assert (y ∈ ran(f)).
```

```
    { rewrite H3; unfold PlusOne; apply Theorem4; right.
      unfold Singleton; apply AxiomII; split; Ens. }
    double H7; rewrite Lemma96' in H8;
    apply Property_Value in H8; auto.
    apply Property_ran in H8; rewrite <- Lemma96 in H8;
    rewrite H2 in H8.
    unfold PlusOne in H8; apply Theorem4 in H8; destruct H8.
    { exists (f⁻¹)[y]; unfold Restriction; apply Theorem4'.
      split; SplitEnsP; split; try apply Theorem19; Ens. }
    { unfold Singleton in H8; apply AxiomII in H8; destruct H8.
      rewrite <- H9 in H5; try apply Theorem19; Ens.
      rewrite <- Lemma96''' in H5; auto.
      generalize (Theorem101 y); intros; contradiction. }
  * unfold Range in H6; apply AxiomII in H6; destruct H6, H8;
    exists x0.
    AssE [x0,z]; unfold Restriction; apply Theorem4'; split.
    { unfold En_g'; apply AxiomII_P; split; auto; left.
      unfold Setminus; apply Theorem4'; split; auto;
      unfold Complement.
      apply AxiomII; split; auto; intro; apply Theorem4 in H10.
      destruct H10; apply AxiomII in H10; destruct H10.
      - assert ([x,f[x]] ∈ 𝒰); clear H10.
        { apply Theorem19;Ens;exists f;apply Property_Value;auto.
          rewrite H2; unfold PlusOne; apply Theorem4; right.
          unfold Singleton; apply AxiomII; split; Ens. }
        apply H11 in H12; clear H11; apply Theorem49 in H9.
        apply Theorem55 in H12; auto; destruct H12; tauto.
      - assert ([(f⁻¹)[y], y] ∈ 𝒰); clear H10.
        { apply Theorem19; Ens; exists f. assert (y ∈ ran(f)).
          { rewrite H3; unfold PlusOne; apply Theorem4; right.
            apply AxiomII; split; Ens. }
          rewrite Lemma96' in H10;apply Property_Value in H10;auto.
          apply AxiomII_P in H10; apply H10. }
        apply H11 in H12; clear H11; apply Theorem49 in H9.
        apply Theorem55 in H12;auto;destruct H12;rewrite H11 in H5.
        generalize (Theorem101 y); intros; contradiction. }
    { double H8; apply Property_dom in H10; rewrite H2 in H10.
      unfold PlusOne in H10;apply Theorem4 in H10;unfold Cartesian.
      apply AxiomII_P;repeat split;auto;try apply Theorem19;Ens.
      destruct H10; auto; apply AxiomII in H10; destruct H10.
      rewrite H11 in H8; try apply Theorem19; Ens; double H8.
      apply Property_dom in H12; apply Property_Value in H12; auto.
      add ([x,z]∈ f)H12; apply H1 in H12; symmetry in H12; tauto. }
Qed.
```

```
Hint Resolve Theorem163 : set.
```

定理 164 $\omega \subset C.$

```
Theorem Theorem164 : ω ⊂ C.
Proof.
  intros.
  unfold Included; apply Mathematical_Induction.
  - assert (∅ ∈ ω); try apply Theorem135; try apply ω.
    unfold ω in H;apply AxiomII in H;destruct H;unfold Integer in H0.
    destruct H0; unfold C; apply AxiomII.
    unfold Cardinal_Number, Ordinal_Number; repeat split; intros; auto.
    + unfold R; apply AxiomII; split; auto.
    + unfold Less in H3; generalize (Theorem16 y); contradiction.
  - intros;destruct H;double H;apply Theorem134 in H1;unfold ω in H1.
    apply AxiomII in H1; unfold Integer in H1; destruct H1, H2.
    unfold C in H0; apply AxiomII in H0; destruct H0.
    unfold Cardinal_Number, Ordinal_Number in H4; destruct H4.
    unfold C; apply AxiomII; split; auto; split; intros.
    + unfold Ordinal_Number, R; apply AxiomII; split; auto.
    + unfold Less, PlusOne in H7; apply Theorem4 in H7; destruct H7.
      * assert (y ∈ ω).
        { unfold ω; apply AxiomII; split; Ens.
          unfold ω in H; apply AxiomII in H; destruct H.
          apply Theorem132 in H7; auto. }
        intro; clear H6; double H8; apply AxiomII in H6; destruct H6.
        unfold Integer in H10; destruct H10; unfold WellOrdered in H11.
        destruct H11 as [H12 H11]; clear H12.
        generalize (classic (y = ∅)); intros; destruct H12.
        { rewrite H12 in H9; clear H12; unfold Equivalent in H9.
          destruct H9 as [f H9]; destruct H9, H12.
          assert (k ∈ (PlusOne k)).
          { unfold PlusOne; apply Theorem4; right; unfold Singleton.
            apply AxiomII; split; Ens. }
          rewrite <- H12 in H14; unfold Function1_1 in H9; destruct H9.
          apply Property_Value in H14; auto; apply Property_ran in H14.
          rewrite H13 in H14; generalize (Theorem16 f[k]);
          contradiction. }
        { assert (y ⊂ y /\ y ≠ ∅). { split; unfold Included; Ens. }
          apply H11 in H13; clear H11 H12; destruct H13.
          assert (y = PlusOne x).
          { apply Theorem133; split; auto; try apply AxiomII; Ens. }
          unfold FirstMember in H11; destruct H11; clear H13.
          rewrite H12 in H9; apply Theorem163 in H9; auto.
          - assert (x ∈ R /\ x ≺ k).
```

```
          { unfold Less; split.
            - unfold R; apply AxiomII; split; Ens.
              apply Theorem111 with (x:= y); auto.
            - unfold R in H4; apply AxiomII in H4; destruct H4.
              unfold Ordinal, full in H13; destruct H13.
              apply H14 in H7; apply H7 in H11; auto. }
          destruct H13; apply H5 in H14; auto.
        - generalize Property_ω; intros; unfold Ordinal, full in H13.
          destruct H13; apply H14 in H8; apply H8 in H11; auto. }
    * unfold Singleton in H7; apply AxiomII in H7; destruct H7.
      assert (k ∈ 𝒰); try apply Theorem19; Ens; apply H8 in H9.
      clear H6 H7 H8; rewrite H9; intro; clear H9; double H.
      apply AxiomII in H7;clear H0;destruct H7;unfold Integer in H7.
      destruct H7; unfold WellOrdered in H8; destruct H8; clear H8.
      generalize (classic (k = ∅)); intros; destruct H8.
      { rewrite H8 in H6; clear H8; unfold Equivalent in H6.
        destruct H6 as [f H6]; destruct H6, H8.
        assert (∅ ∈ (PlusOne ∅)).
        { unfold PlusOne; apply Theorem4; right; unfold Singleton.
          apply AxiomII; split; auto; generalize AxiomVIII; intros.
          destruct H11, H11, H12; Ens. }
        rewrite <- H8 in H11; unfold Function1_1 in H6; destruct H6.
        apply Property_Value in H11; auto; apply Property_ran in H11.
        rewrite H10 in H11; generalize (Theorem16 f[∅]);
        contradiction. }
      { assert (k ⊂ k /\ k ≠ ∅). { split; unfold Included; Ens. }
        apply H9 in H10; clear H8 H9; destruct H10.
        assert (k = PlusOne x).
        { apply Theorem133; split; auto; try apply AxiomII; Ens. }
        unfold FirstMember in H8; destruct H8; clear H10.
        pattern k at 2 in H6; rewrite H9 in H6;
        apply Theorem163 in H6; auto.
        - apply H5 in H8; try contradiction; unfold R; apply AxiomII.
          split; Ens; apply Theorem111 with (x:= k); auto.
        - unfold ω; apply AxiomII; split; Ens.
          apply AxiomII in H; destruct H;
          apply Theorem132 in H8; auto. }
Qed.

Hint Resolve Theorem164 : set.
```

定理 165 $\omega \in C$.

```
Theorem Theorem165 : ω ∈ C.
Proof.
```

```
  generalize Theorem138; intros; AssE ω.
  unfold C; apply AxiomII; split; auto.
  unfold Cardinal_Number; split; intros.
  - unfold Ordinal_Number; auto.
  - unfold Less in H2; intro; double H2.
    apply Theorem134 in H4.
    assert (Ensemble (PlusOne y) /\ y ⊂ (PlusOne y)).
    { split; Ens; unfold PlusOne, Included; intros.
      apply Theorem4; tauto. }
    apply Theorem158 in H5.
    assert (Ensemble ω /\ (PlusOne y) ⊂ ω).
    { split; auto; unfold PlusOne, Included; intros.
      apply Theorem4 in H6; destruct H6.
      - unfold ω in H2; apply AxiomII in H2; destruct H2.
        unfold ω; apply AxiomII; split; Ens.
        apply Theorem132 in H6; auto.
      - unfold Singleton in H6; apply AxiomII in H6; destruct H6.
        rewrite H7; try apply Theorem19; Ens. }
    apply Theorem158 in H6; apply Theorem154 in H3; Ens.
    unfold LessEqual in H5, H6; destruct H5, H6.
    + generalize (Theorem102 P[y] P[PlusOne y]); intros.
      rewrite H3 in H6; elim H7; split; auto.
    + rewrite H3 in H6; rewrite <- H6 in H5.
      generalize (Theorem101 P[PlusOne y]); intros; contradiction.
    + rewrite H3, H5 in H6.
      generalize (Theorem101 P[PlusOne y]); intros; contradiction.
    + apply Theorem154 in H5; Ens.
      apply Theorem164 in H4; unfold C in H4.
      apply AxiomII in H4; destruct H4; unfold Cardinal_Number in H7.
      destruct H7; apply H8 in H1.
      * elim H1; apply Theorem146; auto.
      * unfold Less, PlusOne; apply Theorem4; right.
        unfold Singleton; apply AxiomII; Ens.
Qed.

Hint Resolve Theorem165 : set.
```

定义 166 x 是有限的 $\Longleftrightarrow$ $P(x) \in \omega$.

```
Definition Finite x : Prop := P [x] ∈ ω.

Corollary Property_Finite : forall x, Finite x -> Ensemble x.
Proof.
  intros; unfold Finite in H.
  generalize (classic (Ensemble x)); intros; destruct H0; auto.
```

```
  generalize Theorem152; intros; destruct H1, H2.
  assert (x ∉ dom(P)).
  { rewrite H2; intro; apply Theorem19 in H4; contradiction. }
  apply Theorem69 in H4; rewrite H4 in H; AssE 𝒰.
  generalize Theorem39; intros; contradiction.
Qed.

Hint Unfold Finite : set.
```

定理 167　x 是有限的 $\Longleftrightarrow$ $(\exists\, r,\ r$ 良序 $x,\ r^{-1}$ 也良序 $x)$.

```
Lemma Lemma167 : ∀ r x f,
  WellOrdered r P[x] -> Function1_1 f -> dom(f) = x ->
  ran(f) = P[x] -> WellOrdered \{\ λ u v, Rrelation f[u] r f[v] \}\ x.
Proof.
  intros.
  unfold Function1_1 in H0; destruct H0.
  unfold WellOrdered; split; intros.
  - unfold Connect; intros; destruct H4; rewrite <- H1 in H4, H5.
    AssE u; AssE v; apply Property_Value in H4; auto.
    apply Property_Value in H5; auto; double H4; double H5.
    apply Property_ran in H8; apply Property_ran in H9.
    rewrite H2 in H8, H9; unfold WellOrdered, Connect in H.
    destruct H; clear H10; add (f[v] ∈ P[x]) H8; apply H in H8.
    clear H H9; destruct H8 as [H | [H | H]].
    + left; unfold Rrelation; apply AxiomII_P; split;
      try apply Theorem49; auto.
    + right; left; apply AxiomII_P; split; try apply Theorem49; auto.
    + right; right; rewrite H in H4; clear H.
      assert ([f[v],u] ∈ f⁻¹ /\ [f[v],v] ∈ f⁻¹).
      { unfold Inverse; split; apply AxiomII_P; split; auto.
        - apply Theorem49; split; apply Property_ran in H4; Ens.
        - apply Theorem49; split; apply Property_ran in H5; Ens. }
      unfold Function in H3; apply H3 in H; auto.
  - assert (ran(f|(y)) ⊂ P [x] /\ ran(f|(y)) ≠ ∅).
    { destruct H4; split.
      - unfold Included; intros;unfold Range in H6;apply AxiomII in H6.
        destruct H6, H7;unfold Restriction in H7;apply Theorem4' in H7.
        destruct H7; apply Property_ran in H7; rewrite H2 in H7; auto.
      - apply Property_NotEmpty in H5; destruct H5; double H5;
        apply H4 in H6.
        rewrite <- H1 in H6; apply Property_Value in H6; auto.
        double H6; apply Property_ran in H7; apply Property_NotEmpty.
        exists f[x0]; unfold Range; apply AxiomII; split; Ens.
        exists x0; unfold Restriction; apply Theorem4'; split; auto.
```

```
        unfold Cartesian; apply AxiomII_P; repeat split; Ens.
        apply Theorem19; Ens. }
    apply H in H5; unfold FirstMember in H5; destruct H5, H5.
    unfold Range in H5; apply AxiomII in H5; destruct H5, H7.
    unfold Restriction in H7; apply Theorem4' in H7; destruct H7.
    exists x1; unfold FirstMember; split; intros.
    + unfold Cartesian in H8; apply AxiomII_P in H8; apply H8.
    + clear H8; double H9; apply H4 in H9; rewrite <- H1 in H9.
      apply Property_Value in H9; auto.
      assert (f[y0] ∈ ran(f|(y))).
      { AssE [y0,f[y0]]; apply Theorem49 in H10; destruct H10.
        unfold Range; apply AxiomII; split; auto.
        exists y0; unfold Restriction; apply Theorem4'; split; auto.
        apply AxiomII_P; repeat split; try apply Theorem49; auto.
        apply Theorem19; auto. }
      apply H6 in H10; clear H6; intro; elim H10; clear H10.
      unfold Rrelation at 1 in H6; apply AxiomII_P in H6; destruct H6.
      double H7; apply Property_dom in H11;
      apply Property_Value in H11; auto.
      add ([x1,f[x1]] ∈ f) H7; apply H0 in H7; rewrite H7; auto.
Qed.

Theorem Theorem167 : ∀ x,
  Finite x <-> exists r, WellOrdered r x /\ WellOrdered (r⁻¹) x.
Proof.
  intros; split; intros.
  - double H; unfold Finite in H; apply Property_Finite in H0.
    unfold ω in H; apply AxiomII in H; destruct H.
    unfold Integer in H1; destruct H1; apply Theorem107 in H1.
    apply Theorem153 in H0; apply Theorem146 in H0.
    unfold Equivalent in H0; destruct H0 as [f H0], H0, H3.
    exists (\{\ λ u v, Rrelation f[u] E f[v] \}\); split.
    + apply Lemma167; auto.
    + assert (\{\ λ u v, Rrelation f [u] E f [v] \}\ ⁻¹
             = \{\ λ u v, Rrelation f [u] E⁻¹ f [v] \}\).
      { apply AxiomI; split; intros.
        - PP H5 a b;apply AxiomII_P;apply AxiomII_P in H6;destruct H6.
          apply AxiomII_P in H7; destruct H7; split; auto.
          unfold Rrelation in H8; unfold Rrelation, Inverse.
          apply AxiomII_P; split; auto; AssE [f[b],f[a]].
          apply Theorem49 in H9; destruct H9; apply Theorem49; auto.
        - PP H5 a b; apply AxiomII_P in H6; destruct H6.
          unfold Rrelation, Inverse in H7; apply AxiomII_P in H7;
          destruct H7.
          apply Theorem49 in H6; destruct H6; apply AxiomII_P.
```

```
        split; try apply Theorem49; auto; apply AxiomII_P.
        split; try apply Theorem49; auto. }
    rewrite H5; apply Lemma167; auto.
- destruct H as [r H], H; unfold Finite.
  generalize Theorem113; intros; destruct H1; clear H2.
  apply Theorem107 in H1; add (WellOrdered E R) H; clear H1.
  apply Theorem99 in H; destruct H as [f H], H, H1.
  unfold Order_PXY in H1; destruct H1, H3, H4, H5; double H6.
  apply Theorem114 in H7; add (Ordinal ω) H7; try apply Property_ω.
  apply Theorem110 in H7; destruct H7.
  + destruct H2.
    * assert (P[x] = ran(f)).
      { apply Theorem164 in H7; clear H0; AssE ran(f).
        apply Theorem96 in H4; destruct H4; clear H8.
        assert (dom(f) ≈ ran(f)).{unfold Equivalent; exists f; auto.}
        unfold Function1_1 in H4; destruct H4; rewrite (Lemma96 f),
        (Lemma96' f) in *.
        apply AxiomV in H0; auto; rewrite H2 in *; double H0.
        apply Theorem153 in H0; apply Property_PClass in H10.
        apply Theorem147 with (z:= dom(f⁻¹)) in H0; auto; clear H2 H8.
        unfold C in H7, H10;apply AxiomII in H7;apply AxiomII in H10.
        destruct H7, H10; clear H2 H8;
        unfold Cardinal_Number in H7, H10.
        destruct H7, H10; unfold Ordinal_Number in H2, H8;
        double H2; double H8.
        unfold R in H11, H12; apply AxiomII in H11;
        apply AxiomII in H12.
        destruct H11, H12; clear H11 H12;
        add (Ordinal dom(f⁻¹)) H14; clear H13.
        apply Theorem110 in H14;
        destruct H14 as [H11 | [H11 | H11]]; auto.
        - apply H7 in H11; auto;apply Theorem146 in H0;contradiction.
        - apply H10 in H11; auto; contradiction. }
        rewrite H8; auto.
    * rewrite H2 in H7; add (ω ∈ R) H7; try apply Theorem138.
      generalize (Theorem102 R ω); intros; contradiction.
  + assert (ω ⊂ ran(f)).
    { destruct H7; try (rewrite H7; unfold Included; auto).
      apply Theorem114 in H6; unfold Ordinal, full in H6.
      destruct H6; apply H8 in H7; auto. }
    assert (~ exists z, FirstMember z E⁻¹ ω).
    { intro; destruct H9; unfold FirstMember in H9; destruct H9.
      AssE x0; apply Theorem134 in H9; AssE (PlusOne x0).
      apply H10 in H9; elim H9; clear H9 H10.
      unfold Rrelation, Inverse, E; apply AxiomII_P.
```

```
  split; try apply Theorem49; auto; apply AxiomII_P.
  split; try apply Theorem49; auto; unfold PlusOne.
  apply Theorem4; right; apply AxiomII; auto. }
double H5; unfold Section in H10; destruct H10; clear H11.
apply Lemma97 with (r:= r⁻¹) in H10; auto; clear H6; double H4.
apply Theorem96 in H6; destruct H6; clear H11;
destruct H6 as [H11 H6].
clear H11; elim H9; clear H9; unfold WellOrdered in H10;
destruct H10.
assert (ran(f⁻¹|(ω)) ⊂ dom(f) /\ ran(f⁻¹|(ω)) ≠ ∅).
{ split; unfold Included; intros.
  - unfold Range in H11; apply AxiomII in H11; destruct H11, H12.
    unfold Restriction in H12;apply Theorem4' in H12;destruct H12.
    unfold Inverse in H12; apply AxiomII_P in H12; destruct H12.
    apply Property_dom in H14; auto.
  - assert (∅ ∈ ω);try apply Theorem135;auto;double H11;
    apply H8 in H12.
    rewrite Lemma96' in H12; apply Property_Value in H12; auto.
    AssE [∅,(f⁻¹)[∅]]; apply Theorem49 in H13;
    destruct H13; apply Property_NotEmpty.
    exists f⁻¹[∅]; unfold Range; apply AxiomII; split; auto.
    exists ∅; unfold Restriction; apply Theorem4'; split; auto.
    apply AxiomII_P; repeat split; try apply Theorem49; auto.
    apply Theorem19; auto. }
apply H10 in H11; clear H10; destruct H11; exists f[x0].
unfold FirstMember in H10; destruct H10;
unfold FirstMember; split; intros.
* clear H11; unfold Range in H10; apply AxiomII in H10;
  destruct H10, H11.
  unfold Restriction in H11;apply Theorem4' in H11;destruct H11.
  apply AxiomII_P in H12; destruct H12; clear H12;
  destruct H13; clear H13.
  unfold Inverse in H11; apply AxiomII_P in H11; destruct H11;
  double H13.
  apply Property_dom in H14; apply Property_Value in H14; auto.
  add ([x0,f[x0]] ∈ f) H13; clear H14;
  unfold Function in H; apply H in H13.
  rewrite H13 in H12; auto.
* double H12; apply H8 in H13; apply AxiomII in H13;
  destruct H13, H14.
  AssE [x1,y]; apply Theorem49 in H15; destruct H15; clear H16.
  assert (x1 ∈ ran(f⁻¹|(ω))).
  { unfold Range; apply AxiomII; split; auto; exists y.
    unfold Restriction; apply Theorem4'; split.
    - unfold Inverse; apply AxiomII_P; split;
```

```
          try apply Theorem49; auto.
        - unfold Cartesian; apply AxiomII_P; repeat split;
          try apply Theorem49; auto.
          apply Theorem19; auto. }
      apply H11 in H16; clear H11; unfold Range in H10;
      apply AxiomII in H10.
      destruct H10, H11; unfold Restriction in H11;
      apply Theorem4' in H11.
      destruct H11; clear H17; unfold Inverse in H11;
      apply AxiomII_P in H11.
      clear H10; destruct H11; apply Property_dom in H11; double H14.
      apply Property_dom in H17; add (x1 ∈ dom(f)) H11;
      double H11; clear H17.
      unfold Connect in H9; apply H9 in H18; clear H9; intro.
      unfold Rrelation, Inverse, E in H9;
      apply AxiomII_P in H9; destruct H9.
      clear H9; apply AxiomII_P in H17; destruct H17.
      destruct H18 as [H18|[H18|H18]]; try contradiction.
      { clear H16; unfold Order_Pr in H4; destruct H11.
        assert (x1 ∈ dom(f) /\ x0 ∈ dom(f) /\ Rrelation x1 r x0).
        { repeat split; auto; unfold Rrelation, Inverse in H18.
          apply AxiomII_P in H18; unfold Rrelation; apply H18. }
        apply H4 in H19; clear H4 H9 H13 H15;
        unfold Rrelation, E in H19.
        apply AxiomII_P in H19; destruct H19; clear H4.
        apply Property_Value in H16; auto; add ([x1,f[x1]] ∈ f) H14.
        apply H in H14; rewrite H14 in H17; add (f[x1] ∈ f[x0]) H17.
        generalize (Theorem102 f[x0] f[x1]); intros; contradiction. }
      { rewrite H18 in H17; clear H9 H15 H16; destruct H11.
        apply Property_Value in H11; auto; add ([x1,y] ∈ f) H11.
        unfold Function in H; apply H in H11; rewrite H11 in H17.
        generalize (Theorem101 y); intros; contradiction. }
Qed.

Hint Resolve Theorem167 : set.

(* Some properties about finite *)

Lemma Finite_Included : ∀ x y,
  Finite x -> y ⊂ x -> Finite y.
Proof.
  intros.
  apply Theorem167 in H; destruct H as [r H], H.
  apply Theorem167; exists r; split.
  - unfold WellOrdered, Connect in H; destruct H.
```

```
    unfold WellOrdered, Connect; split; intros.
    + destruct H3; apply H; auto.
    + destruct H3; apply H2; split; auto.
      add (y ⊂ x) H3; apply Theorem28 in H3; auto.
  - unfold WellOrdered, Connect in H1; destruct H1.
    unfold WellOrdered, Connect; split; intros.
    + destruct H3; apply H1; auto.
    + destruct H3; apply H2; split; auto.
      add (y ⊂ x) H3; apply Theorem28 in H3; auto.
Qed.

Hint Resolve Finite_Included : set.

Lemma Finite_Single : ∀ z, Ensemble z -> Finite ([z]).
Proof.
  intros.
  apply Theorem167; exists E; split.
  - unfold WellOrdered; split; intros.
    + unfold Connect; intros; destruct H0; unfold Singleton in H0, H1.
      apply AxiomII in H0; apply AxiomII in H1;
      destruct H0, H1; double H.
      apply Theorem19 in H; apply Theorem19 in H4; apply H2 in H.
      apply H3 in H4; rewrite <- H4 in H; tauto.
    + destruct H0; apply Property_NotEmpty in H1; destruct H1; exists x.
      unfold FirstMember; split; auto; intros; unfold Included in H0.
      apply H0 in H1; apply H0 in H2;
      unfold Singleton in H1, H2; double H.
      apply AxiomII in H1; apply AxiomII in H2; destruct H1, H2.
      apply Theorem19 in H; apply Theorem19 in H3; apply H4 in H.
      apply H5 in H3; rewrite <- H3 in H; rewrite H.
      intro; unfold Rrelation in H6; unfold E in H6;
      apply AxiomII_P in H6.
      destruct H6; generalize (Theorem101 y0); intros; contradiction.
  - unfold WellOrdered; split; intros.
    + unfold Connect; intros; destruct H0; unfold Singleton in H0, H1.
      apply AxiomII in H0; apply AxiomII in H1;
      destruct H0, H1; double H.
      apply Theorem19 in H; apply Theorem19 in H4.
      apply H2 in H; apply H3 in H4; rewrite <- H4 in H; tauto.
    + destruct H0; apply Property_NotEmpty in H1; auto;
      destruct H1; exists x.
      unfold FirstMember; split; auto; intros; unfold Included in H0.
      apply H0 in H1; apply H0 in H2;
      unfold Singleton in H1, H2; double H.
      apply AxiomII in H1; apply AxiomII in H2; destruct H1, H2.
```

```
    apply Theorem19 in H; apply Theorem19 in H3; apply H4 in H.
    apply H5 in H3; rewrite <- H3 in H; rewrite H.
    intro; unfold Rrelation in H6; unfold Inverse in H6.
    apply AxiomII_P in H6; destruct H6; unfold E in H7;
    apply AxiomII_P in H7.
    destruct H7; generalize (Theorem101 y0); intros; contradiction.
Qed.

Hint Resolve Finite_Single : set.
```

下面这些关于 "**有限集**"的定理都能对集的势进行归纳证明, 或者用构造一个良序再应用定理 167 加以证明.

定理 168　*x 和 y 均有限 $\Longrightarrow$ $x \cup y$ 也是有限的.*

```
Theorem Theorem168 : ∀ x y,
  Finite x /\ Finite y -> Finite (x ∪ y).
Proof.
  intros; destruct H.
  apply Theorem167 in H; apply Theorem167 in H0.
  destruct H as [r H], H0 as [s H0], H, H0; apply Theorem167.
  exists (\{\ λ u v, (u ∈ x /\ v ∈ x /\ Rrelation u r v) \/
  (u∈(y~x) /\ v∈(y~x) /\ Rrelation u s v) \/ (u∈x /\ v∈(y~x)) \}\);
  split.
  - clear H1 H2; unfold WellOrdered in H, H0; destruct H, H0.
    unfold WellOrdered; split; intros.
    + clear H1 H2; unfold Connect in H, H0; unfold Connect; intros.
      destruct H1; apply Theorem4 in H1; apply Theorem4 in H2.
      unfold Rrelation; destruct H1, H2.
      * clear H0; assert (u ∈ x /\ v ∈ x); auto; apply H in H0.
        clear H; destruct H0 as [H | [H | H]]; try tauto.
        { left; SplitEnsP. } { right; left; SplitEnsP. }
      * clear H0; generalize (classic (v ∈ x)); intros; destruct H0.
        { assert (u ∈ x /\ v ∈ x); auto; apply H in H3.
          clear H; destruct H3 as [H | [H | H]]; try tauto.
          - left; SplitEnsP.
          - right; left; SplitEnsP. }
        { left; SplitEnsP; right; right; split; auto; apply Theorem4'.
          split; auto; unfold Complement; SplitEns. }
      * clear H0; generalize (classic (u ∈ x)); intros; destruct H0.
        { assert (u ∈ x /\ v ∈ x); auto; apply H in H3.
          clear H; destruct H3 as [H | [H | H]]; try tauto.
          - left; SplitEnsP.
          - right; left; SplitEnsP. }
        { right; left; SplitEnsP.
          right; right; split; auto; unfold Setminus; apply Theorem4'.
```

```
      split; auto; unfold Complement; SplitEns. }
    * generalize (classic (u∈x)) (classic (v∈x)); intros;
      destruct H3, H4.
      { clear H0; assert (u ∈ x /\ v ∈ x); auto; apply H in H0.
        clear H; destruct H0 as [H | [H | H]]; try tauto.
        - left; SplitEnsP.
        - right; left; SplitEnsP. }
      { left; SplitEnsP; right; right; split; auto; apply Theorem4'.
        split; auto; unfold Complement; SplitEns. }
      { right; left; SplitEnsP; right; right;
        split; auto; apply Theorem4'.
        split; auto; unfold Complement; SplitEns. }
      { clear H; assert (u ∈ y /\ v ∈ y); auto; apply H0 in H.
        clear H0; destruct H as [H | [H | H]]; try tauto.
        - left; SplitEnsP; right; left; repeat split; auto.
          + apply Theorem4'; split; auto; SplitEns.
          + apply Theorem4'; split; auto; SplitEns.
        - right; left; SplitEnsP.
          right; left; unfold Setminus; repeat split; auto.
          + apply Theorem4'; split; auto; SplitEns.
          + apply Theorem4'; split; auto; SplitEns. }
  + generalize (classic (\{ λ z, z ∈ y0 /\ z ∈ x \} = ∅)).
    clear H H0; destruct H3; intros; destruct H3.
    * assert (y0 ⊂ y).
      { unfold Included; intros; double H4.
        apply H in H5; apply Theorem4 in H5; destruct H5; auto.
        generalize (Theorem16 z); intros; elim H6; clear H6.
        rewrite <- H3; apply AxiomII; repeat split; Ens. }
      add (y0 ≠ ∅) H4; apply H2 in H4; clear H0 H1 H2.
      destruct H4 as [z H0]; exists z; unfold FirstMember in H0.
      destruct H0; unfold FirstMember; split; auto; intros.
      double H2; apply H1 in H4; clear H1; intro; elim H4; clear H4.
      unfold Rrelation in H1; apply AxiomII_P in H1; destruct H1.
      unfold Rrelation; destruct H4 as [H4|[H4|H4]]; try apply H4.
      { destruct H4; clear H5; generalize (Theorem16 y1); intros.
        elim H5; rewrite <- H3; apply AxiomII; repeat split; Ens. }
      { destruct H4; clear H5; generalize (Theorem16 y1); intros.
        elim H5; rewrite <- H3; apply AxiomII; repeat split; Ens. }
    * assert (\{λ z, z∈y0 /\ z∈x\} ⊂ x).
      { unfold Included; intros; apply AxiomII in H4; apply H4. }
      add (\{λ z, z∈y0 /\ z∈x\} <> ∅) H4; apply H1 in H4; clear H1 H2.
      destruct H4 as [z H1]; exists z; unfold FirstMember in H1.
      destruct H1; apply AxiomII in H1; destruct H1, H4.
      unfold FirstMember; split; auto; intros.
      generalize (classic (y1∈x)); intros; destruct H7.
```

```
    { assert (y1 ∈ \{λ z, z∈y0 /\ z∈x\}).
      { apply AxiomII; repeat split; Ens. }
      apply H2 in H8; intro; elim H8; clear H2 H8.
      unfold Rrelation in H9; apply AxiomII_P in H9; destruct H9.
      unfold Rrelation; destruct H8 as [H8|[H8|H8]]; try apply H8.
      - destruct H8; clear H9; unfold Setminus in H8;
        apply AxiomII in H8.
        destruct H8, H9; unfold Complement in H10;
        apply AxiomII in H10.
        destruct H10; contradiction.
      - destruct H8; clear H8; unfold Setminus in H9;
        apply AxiomII in H9.
        destruct H9, H9; unfold Complement in H10;
        apply AxiomII in H10.
        destruct H10; contradiction. }
    { intro; unfold Rrelation in H8; apply AxiomII_P in H8.
      destruct H8, H9 as [H9|[H9|H9]], H9; try contradiction.
      destruct H10; clear H8 H9 H11; unfold Setminus in H10.
      apply AxiomII in H10; destruct H10, H9;
      unfold Complement in H10.
      apply AxiomII in H10; destruct H10; contradiction. }
- unfold WellOrdered; split; intros.
  + clear H1 H2; unfold WellOrdered in H, H0; destruct H, H0.
    clear H1 H2; unfold Connect in H, H0; unfold Connect; intros.
    destruct H1; apply Theorem4 in H1; apply Theorem4 in H2.
    unfold Rrelation; destruct H1, H2.
    * clear H0; assert (u ∈ x /\ v ∈ x); auto; apply H in H0.
      clear H; destruct H0 as [H | [H | H]]; try tauto.
      { right; left; unfold Inverse; SplitEnsP; SplitEnsP. }
      { left; SplitEnsP; SplitEnsP. }
    * clear H0; generalize (classic (v ∈ x)); intros; destruct H0.
      { assert (u ∈ x /\ v ∈ x); auto; apply H in H3.
        clear H; destruct H3 as [H | [H | H]]; try tauto.
        - right; left; SplitEnsP; SplitEnsP.
        - left; SplitEnsP; SplitEnsP. }
      { right; left; SplitEnsP; SplitEnsP.
        right; right; split; auto; apply Theorem4'.
        split; auto; unfold Complement; SplitEns. }
    * clear H0; generalize (classic (u ∈ x)); intros; destruct H0.
      { assert (u ∈ x /\ v ∈ x); auto; apply H in H3.
        clear H; destruct H3 as [H | [H | H]]; try tauto.
        - right; left; SplitEnsP; SplitEnsP.
        - left; SplitEnsP; SplitEnsP. }
      { left; SplitEnsP; SplitEnsP.
        right; right; split; auto; unfold Setminus; apply Theorem4'.
```

```
      split; auto; unfold Complement; SplitEns. }
    * generalize (classic (u∈x)) (classic (v∈x)); intros;
      destruct H3, H4.
      { clear H0; assert (u ∈ x /\ v ∈ x); auto; apply H in H0.
        clear H; destruct H0 as [H | [H | H]]; try tauto.
        - right; left; SplitEnsP; SplitEnsP.
        - left; SplitEnsP; SplitEnsP. }
      { right; left; SplitEnsP; SplitEnsP.
        right; right; split; auto; apply Theorem4'.
        split; auto; unfold Complement; SplitEns. }
      { left; SplitEnsP; SplitEnsP.
        right; right; split; auto; unfold Setminus; apply Theorem4'.
        split; auto; unfold Complement; SplitEns. }
      { clear H; assert (u ∈ y /\ v ∈ y); auto; apply H0 in H.
        clear H0; destruct H as [H | [H | H]]; try tauto.
        - right; left; SplitEnsP; SplitEnsP.
          right; left; unfold Setminus; repeat split; auto.
          + apply Theorem4'; split; auto; SplitEns.
          + apply Theorem4'; split; auto; SplitEns.
        - left; SplitEnsP; SplitEnsP; right; left; repeat split; auto.
          + apply Theorem4'; split; auto; SplitEns.
          + apply Theorem4'; split; auto; SplitEns. }
  + clear H H0; unfold WellOrdered in H1, H2.
    destruct H1, H2; clear H H1; destruct H3.
    generalize (classic (\{λ z, z∈y0 /\ z∈(y∼x)\}=∅)); intros;
    destruct H3.
    * assert (y0 ⊂ x).
      { unfold Included; intros; double H4.
        apply H in H5; apply Theorem4 in H5; destruct H5; auto.
        generalize (classic (z ∈ x)); intros; destruct H6; auto.
        generalize (Theorem16 z); intros; elim H7; clear H7.
        rewrite <- H3; apply AxiomII; repeat split; Ens.
        unfold Setminus; apply Theorem4'; split; auto.
        unfold Complement; apply AxiomII; split; Ens. }
      add (y0 ≠ ∅) H4; apply H0 in H4; clear H0 H1 H2.
      destruct H4 as [z H0]; exists z; unfold FirstMember in H0.
      destruct H0; unfold FirstMember; split; auto; intros.
      double H2; apply H1 in H4; clear H1; intro; elim H4; clear H4.
      unfold Rrelation in H1; apply AxiomII_P in H1; destruct H1.
      apply AxiomII_P in H4; destruct H4 as [H5 H4]; clear H5.
      unfold Rrelation, Inverse; apply AxiomII_P; split; auto.
      destruct H4 as [H4|[H4|H4]]; try apply H4.
      { destruct H4; clear H5; generalize (Theorem16 z); intros.
        elim H5; rewrite <- H3; apply AxiomII; repeat split; Ens. }
      { destruct H4; clear H4; generalize (Theorem16 y1); intros.
```

```
        elim H4; rewrite <- H3; apply AxiomII; repeat split; Ens. }
    * assert (\{λ z, z∈y0 /\ z∈(y~x)\} ⊂ y).
      { unfold Included; intros; apply AxiomII in H4; destruct H4, H5.
        unfold Setminus in H6; apply Theorem4' in H6; apply H6. }
      add (\{λ z, z∈y0 /\ z∈(y~x)\} <> ∅) H4; apply H2 in H4;
      clear H0 H2.
      destruct H4 as [z H0]; exists z; unfold FirstMember in H0.
      destruct H0; apply AxiomII in H0; destruct H0, H4.
      unfold Setminus in H5; apply Theorem4' in H5; destruct H5.
      unfold Complement in H6; apply AxiomII in H6; clear H0;
      destruct H6.
      unfold FirstMember; split; auto; intros.
      generalize (classic (y1∈x)); intros; destruct H8.
      { intro; unfold Rrelation in H9; apply AxiomII_P in H9;
        destruct H9.
        apply AxiomII_P in H10; destruct H10 as [H11 H10];
        clear H11.
        destruct H10 as [H10|[H10|H10]], H10; try contradiction.
        destruct H11; clear H9 H10 H12; unfold Setminus in H11.
        apply AxiomII in H11; destruct H11, H10;
        unfold Complement in H11.
        apply AxiomII in H11; destruct H11; contradiction. }
      { assert (y1 ∈ \{λ z, z ∈ y0 /\ z ∈ (y ~ x)\}).
        { apply AxiomII; repeat split; Ens; apply H in H7.
          apply Theorem4 in H7; destruct H7; try contradiction.
          apply Theorem4'; split; auto; apply AxiomII; split; Ens. }
        apply H2 in H9; intro; elim H9; clear H2 H9.
        unfold Rrelation in H10; apply AxiomII_P in H10; destruct H10.
        apply AxiomII_P in H9; destruct H9 as [H10 H9]; clear H10.
        unfold Rrelation, Inverse; SplitEnsP.
        destruct H9 as [H9|[H9|H9]], H9; try contradiction;
        apply H10. }
Qed.

Hint Resolve Theorem168 : set.
```

定理 169 x 有限且 x 的每个元有限 $\Longrightarrow$ $\bigcup x$ 也有限.

```
Lemma Lemma169 : ∀ x y, x ∈ y -> y = (y ~ [x] ∪ [x]).
Proof.
  intros.
  apply AxiomI; split; intros.
  - generalize (classic (z ∈ [x])); intros; destruct H1.
    + apply Theorem4; right; auto.
    + apply Theorem4; left; unfold Setminus; apply Theorem4'.
```

```
    split; auto; unfold Complement; apply AxiomII; Ens.
  - apply Theorem4 in H0; destruct H0.
    + unfold Setminus in H0; apply Theorem4' in H0; apply H0.
    + unfold Singleton in H0; apply AxiomII in H0; destruct H0.
      rewrite H1; try apply Theorem19; Ens.
Qed.

Lemma Lemma169' : ∀ x y, ⋃(x ∪ y) = (⋃x) ∪ (⋃y).
Proof.
  intros.
  apply AxiomI; split; intros.
  - apply AxiomII in H; destruct H, H0, H0.
    apply Theorem4 in H1; destruct H1.
    + apply Theorem4; left; apply AxiomII; Ens.
    + apply Theorem4; right; apply AxiomII; Ens.
  - apply Theorem4 in H; destruct H.
    + apply AxiomII in H; destruct H, H0, H0.
      apply AxiomII; split; Ens; exists x0.
      split; auto; apply Theorem4; auto.
    + apply AxiomII in H; destruct H, H0, H0.
      apply AxiomII; split; auto; exists x0.
      split; auto; apply Theorem4; auto.
Qed.

Theorem Theorem169 : ∀ (x: Class),
  Finite x -> (∀ z, z∈x -> Finite z) -> Finite (⋃ x).
Proof.
  intros; double H.
  unfold Finite in H; apply Property_Finite in H1.
  assert (\{λ u, u∈ ω /\ (∀ y, P[y] = u /\ Ensemble y /\
          (∀ z, z∈y -> Finite z) -> Finite (⋃ y)) \} = ω).
  { apply Theorem137.
    - unfold Included; intros; apply AxiomII in H2; apply H2.
    - apply AxiomII; generalize (Theorem135 x); intros; destruct H2.
      clear H3; repeat split; Ens; intros; destruct H3, H4.
      generalize (classic (y = ∅)); intros; destruct H6.
      + rewrite H6 in *; rewrite Theorem24'; unfold Finite;
        rewrite H3; auto.
      + apply Property_NotEmpty in H6; destruct H6;
        apply Theorem153 in H1.
        apply Theorem153 in H4; apply Theorem146 in H4;
        unfold Equivalent in H4.
        destruct H4 as [f H4], H4, H4, H7; rewrite <- H7 in H6.
        apply Property_Value in H6; auto; apply Property_ran in H6.
        rewrite H9, H3 in H6; generalize (Theorem16 f[x0]);
```

```
    contradiction.
  - intros; apply AxiomII in H2; apply AxiomII; destruct H2, H3;
    double H3.
    apply Theorem134 in H5; repeat split; Ens; intros;
    destruct H6, H7.
    AssE y; clear H7; double H9; unfold PlusOne in H6;
    apply Theorem153 in H9.
    unfold Equivalent in H9; destruct H9 as [f H9], H9, H9, H10.
    assert (u ∈ P[y]).
    { rewrite H6; apply Theorem4; right; unfold Singleton.
      apply AxiomII; split; Ens. }
    rewrite <- H10 in H13; apply Property_Value in H13; auto.
    apply Property_ran in H13; rewrite H12 in H13.
    apply Lemma169 in H13; rewrite H13; clear H13.
    rewrite Lemma169'; apply Theorem168; split.
    + apply H4; assert (Ensemble (y ~ [f[u]])).
      { apply (Theorem33 y _); auto; unfold Included; intros.
        unfold Setminus in H13; apply Theorem4' in H13; apply H13. }
      repeat split; auto; intros.
      * apply Theorem164 in H3; apply Theorem156 in H3; clear H2;
        destruct H3.
        rewrite <- H3 at 2; add (Ensemble u) H13;
        apply Theorem154 in H13.
        apply H13; clear H13; apply Theorem146; unfold Equivalent.
        exists (f|(P[y]~[u])).
        { repeat split; unfold Relation; intros.
          - unfold Restriction in H13; apply Theorem4' in H13.
            destruct H13; PP H14 a b; Ens.
          - unfold Restriction in H13; destruct H13;
            apply Theorem4' in H13.
            apply Theorem4' in H14; destruct H13, H14;
            unfold Function in H9.
            apply H9 with (x:= x0); split; auto.
          - PP H13 a b; Ens.
          - unfold Inverse, Restriction in H13; destruct H13.
            apply AxiomII_P in H13; apply AxiomII_P in H14;
            destruct H13, H14.
            apply Theorem4' in H15; apply Theorem4' in H16;
            destruct H15, H16.
            clear H17 H18; unfold Function in H11;
            apply H11 with (x:= x0).
            split; apply AxiomII_P; Ens.
          - apply AxiomI; split; intros.
            + unfold Domain in H13; apply AxiomII in H13;
              destruct H13, H14.
```

```
      unfold Restriction in H14; apply Theorem4' in H14;
      destruct H14.
      clear H14; unfold Cartesian in H15;
      apply AxiomII_P in H15.
      destruct H15, H15; clear H16; unfold Setminus in H15.
      apply Theorem4' in H15; destruct H15; rewrite H6 in H15.
      apply Theorem4 in H15; destruct H15; auto.
      unfold Complement in H16; apply AxiomII in H16;
      destruct H16.
      contradiction.
    + unfold Domain; apply AxiomII; split; Ens; exists f[z].
      unfold Restriction; apply Theorem4'.
      assert (z ∈ dom(f)).
      { rewrite H10, H6; apply Theorem4; tauto. }
      apply Property_Value in H14; auto; split; auto.
      unfold Cartesian; apply AxiomII_P; split; Ens;
      double H14.
      apply Property_ran in H15; split;
      try apply Theorem19; Ens.
      clear H15; apply Property_dom in H14; unfold Setminus.
      apply Theorem4'; rewrite H10 in H14; split; auto.
      unfold Complement; apply AxiomII; split; Ens; intro.
      apply AxiomII in H15; destruct H15.
      rewrite H16 in H13; try apply Theorem19; Ens.
      generalize (Theorem101 u); intros; contradiction.
  - apply AxiomI; split; intros.
    + unfold Range in H13; apply AxiomII in H13;
      destruct H13, H14.
      unfold Restriction in H14; apply Theorem4' in H14;
      destruct H14.
      unfold Cartesian in H15; apply AxiomII_P in H15;
      destruct H15.
      clear H15; destruct H16; clear H16;
      unfold Setminus in H15.
      apply Theorem4' in H15; destruct H15; double H15.
      rewrite <- H10 in H15; rewrite H6 in H17.
      apply Theorem4 in H17; destruct H17.
      * clear H17; apply Property_Value in H15; auto.
        add ([x0,z] ∈ f) H15; apply H9 in H15;
        rewrite <- H15 in *.
        clear H15;unfold Setminus; apply Theorem4';double H14.
        apply Property_ran in H15; rewrite H12 in H15;
        split; auto.
        unfold Complement; apply AxiomII; split; Ens; intro.
        apply AxiomII in H17;destruct H17;assert(u ∈ dom(f)).
```

```
              { rewrite H10, H6; apply Theorem4; right.
                unfold Singleton; apply AxiomII; Ens. }
              apply Property_Value in H19; auto; AssE [u,f[u]].
              apply Theorem49 in H20; destruct H20.
              rewrite H18 in H14; try apply Theorem19; Ens.
              unfold Complement in H16; apply AxiomII in H16;
              destruct H16.
              elim H22; unfold Singleton; apply AxiomII;
              split; Ens; intros.
              apply H11 with (x:= f[u]); unfold Inverse.
              split; apply AxiomII_P; split;
              try apply Theorem49; auto.
            * unfold Complement in H16; apply AxiomII in H16.
              destruct H16; contradiction.
          + unfold Setminus in H13; apply Theorem4' in H13;
            destruct H13.
            rewrite <- H12 in H13; apply AxiomII in H13;
            destruct H13, H15.
            unfold Range; apply AxiomII; split; Ens; exists x0.
            unfold Restriction; apply Theorem4'; split; auto.
            unfold Cartesian; apply AxiomII_P; split; Ens.
            split; try apply Theorem19; Ens; apply Theorem4';
            double H15.
            apply Property_dom in H16; rewrite <- H10; split; auto.
            unfold Complement; apply AxiomII; split; Ens; intro.
            apply AxiomII in H17; destruct H17.
            rewrite H18 in *; try apply Theorem19; Ens.
            apply Property_Value in H16; auto; clear H17 H18.
            add ([u,z] ∈ f)H16;apply H9 in H16;rewrite H16 in H14.
            unfold Complement in H14;apply AxiomII in H14;clear H13.
            destruct H14; elim H14; apply AxiomII; Ens. }
        * unfold Setminus in H14; apply Theorem4' in H14; destruct H14.
          apply H8; auto.
      + assert (f[u] ∈ y).
        { assert (u ∈ dom(f)).
          { rewrite H10, H6;apply Theorem4;right;apply AxiomII;Ens. }
          apply Property_Value in H13; auto; apply Property_ran in H13.
          rewrite H12 in H13; auto. }
        AssE f[u]; apply Theorem44 in H14; destruct H14; rewrite H15.
        clear H14 H15; apply H8; auto. }
  rewrite <- H2 in H; clear H2; apply AxiomII in H; destruct H, H2.
  apply H3; repeat split; auto.
Qed.

Hint Resolve Theorem169 : set.
```

定理 170 x 和 y 均有限 $\Longrightarrow$ $x \times y$ 也有限.

```
Theorem Theorem170 : ∀ x y,
  Finite x /\ Finite y -> Finite (x × y).
Proof.
  intros; destruct H.
  generalize (classic (y = ∅)); intros; destruct H1.
  - rewrite H1 in *; clear H1.
    assert ((x × ∅) = ∅).
    { apply AxiomI; split; intros.
      - PP H1 a b; apply AxiomII_P in H2; destruct H2, H3.
        generalize (Theorem16 b); intros; contradiction.
      - generalize (Theorem16 z); intros; contradiction. }
    rewrite H1; auto.
  - assert (⋃ \{ λ u, ∃ v, v∈x /\ u = ([v] × y) \} = x × y).
    { clear H1; apply AxiomI; split; intros.
      - unfold Element_U in H1; apply AxiomII in H1;
        destruct H1, H2, H2.
        apply AxiomII in H3; destruct H3, H4, H4; unfold Cartesian.
        rewrite H5 in H2; PP H2 a b; apply AxiomII_P in H6;
        destruct H6, H7.
        apply AxiomII_P; repeat split; auto; unfold Singleton in H7.
        apply AxiomII in H7; destruct H7; rewrite H9;
        try apply Theorem19; Ens.
      - unfold Cartesian in H1; PP H1 a b; apply AxiomII_P in H2.
        destruct H2, H3; apply AxiomII; split; auto; exists ([a] × y);
        split.
        + unfold Cartesian; apply AxiomII_P; repeat split; auto.
          unfold Singleton; apply AxiomII; split; Ens.
        + apply AxiomII; split; Ens; apply Theorem73; split; Ens.
          apply Property_Finite; auto. }
    rewrite <- H2; clear H2; apply Theorem169; intros.
    + assert (x ≈ \{ λ u, ∃ v, v∈x /\ u=([v] × y) \}).
      { unfold Equivalent; exists (\{\ λ u v, u∈x /\ v=([u] × y)\}\).
        repeat split; intros; try (unfold Relation; intros; PP H2 a b;
        Ens).
        - destruct H2; apply AxiomII_P in H2; apply AxiomII_P in H3.
          destruct H2, H3, H4, H5; rewrite H6, H7; auto.
        - destruct H2; apply AxiomII_P in H2; apply AxiomII_P in H3.
          destruct H2, H3; clear H2 H3; apply AxiomII_P in H4.
          apply AxiomII_P in H5; destruct H4, H5, H3, H5;
          rewrite H7 in H6.
          clear H7; generalize (classic (y0 = z)); intros;
          destruct H7; auto.
          elim H7; clear H7; apply Property_NotEmpty in H1; destruct H1.
```

```
    assert ([y0,x1] ∈ ([z] × y)).
    { rewrite H6; unfold Cartesian; apply AxiomII_P.
      repeat split; try apply Theorem49; Ens.
      unfold Singleton; apply AxiomII; split; Ens. }
    unfold Cartesian in H7;apply AxiomII_P in H7;destruct H7, H8.
    unfold Singleton in H8; apply AxiomII in H8; destruct H8.
    apply H10; apply Theorem19; Ens.
  - apply AxiomI; split; intros.
    + unfold Domain in H2; apply AxiomII in H2; destruct H2, H3.
      apply AxiomII_P in H3; apply H3.
    + unfold Domain; apply AxiomII; split; Ens.
      exists ([z] × y); apply AxiomII_P; repeat split; auto.
      apply Theorem49; split; Ens; apply Property_Finite in H0.
      apply Theorem73; split; Ens.
  - apply AxiomI; split; intros.
    + unfold Range in H2; apply AxiomII in H2; destruct H2, H3.
      apply AxiomII_P in H3;destruct H3, H4;apply AxiomII;split;
      Ens.
    + apply AxiomII in H2; destruct H2, H3, H3.
      unfold Range; apply AxiomII; split; auto; exists x0.
      apply AxiomII_P; repeat split; try apply Theorem49; Ens. }
assert (Ensemble x /\ Ensemble \{λ u, ∃ v, v∈x/\u=([v]×y)\}).
{ apply Property_Finite in H; apply Property_Finite in H0;
  split; auto.
  assert (Ensemble pow(x × y)).
  { apply Theorem38; auto; apply Theorem74; split; auto. }
  apply (Theorem33 pow(x × y) _); auto; unfold Included; intros.
  apply AxiomII in H4; destruct H4, H5, H5;
  rewrite H6 in *; clear H6.
  unfold PowerSet; apply AxiomII; split; auto;
  unfold Included; intros.
  PP H6 a b; apply AxiomII_P in H7; destruct H7, H8;
  apply AxiomII_P.
  repeat split;auto;unfold Singleton in H8;apply AxiomII in H8.
  destruct H8; rewrite H10; try apply Theorem19; Ens. }
apply Theorem154 in H3; apply H3 in H2; clear H3.
unfold Finite; unfold Finite in H; rewrite <- H2; auto.
+ apply AxiomII in H2; destruct H2, H3, H3;
rewrite H4 in *; clear H4.
assert (y ≈ ([x0] × y)).
{ unfold Equivalent; exists (\{\ λ u v, u∈y /\ v = [x0,u] \}\).
  repeat split; intros; try (unfold Relation; intros; PP H4 a b;
  Ens).
  - destruct H4; apply AxiomII_P in H4; apply AxiomII_P in H5.
    destruct H4, H5, H6, H7; rewrite H8, H9; auto.
```

```
    - destruct H4; apply AxiomII_P in H4; apply AxiomII_P in H5.
      destruct H4, H5; clear H4 H5; apply AxiomII_P in H6.
      apply AxiomII_P in H7; destruct H6, H7, H5, H7;
      rewrite H9 in H8.
      apply Theorem55 in H8; destruct H8; Ens.
    - apply AxiomI; split; intros.
      + unfold Domain in H4; apply AxiomII in H4; destruct H4, H5.
        apply AxiomII_P in H5; apply H5.
      + unfold Domain; apply AxiomII; split; Ens.
        exists [x0,z0]; apply AxiomII_P; repeat split; auto.
        apply Theorem49; split; Ens; apply Theorem49; Ens.
    - apply AxiomI; split; intros.
      + unfold Range in H4; apply AxiomII in H4; destruct H4, H5.
        apply AxiomII_P in H5; destruct H5, H6;
        rewrite H7 in *; clear H7.
        apply AxiomII_P; repeat split; auto; unfold Singleton.
        apply AxiomII; split; Ens.
      + PP H4 a b; apply AxiomII_P in H5; destruct H5, H6.
        unfold Range; apply AxiomII; split; auto; exists b.
        apply AxiomII_P; repeat split; try apply Theorem49; Ens.
        unfold Singleton in H6; apply AxiomII in H6; destruct H6.
        rewrite H8; try apply Theorem19; Ens. }
  assert (Ensemble y /\ Ensemble ([x0] × y)).
  { apply Property_Finite in H; apply Property_Finite in H0; Ens. }
  apply Theorem154 in H5; apply H5 in H4; clear H5.
  unfold Finite; unfold Finite in H0; rewrite <- H4; auto.
Qed.

Hint Resolve Theorem170 : set.
```

定理 171 x 有限 $\Longrightarrow$ 2^x 也有限.

```
Lemma Lemma171 : ∀ x y,
  y∈x -> pow(x) = pow(x~[y]) ∪ (\{λ z, z ⊂ x /\ y ∈ z\}).
Proof.
  intros; unfold PowerSet; apply AxiomI.
  split; intros; apply AxiomII in H0; destruct H0; apply AxiomII;
  split; auto.
  - generalize (classic (y ∈ z)); intros; destruct H2.
    + right; apply AxiomII; split; auto.
    + left; apply AxiomII; split; auto;
      unfold Included; intros; double H3.
      unfold Setminus; apply Theorem4'; apply H1 in H4; split; auto.
      unfold Complement; apply AxiomII; split; Ens; intro.
      unfold Singleton in H5; apply AxiomII in H5; destruct H5.
```

```
      rewrite H6 in H3; try contradiction; apply Theorem19; Ens.
  - destruct H1; apply AxiomII in H1; try apply H1;
    clear H0; destruct H1.
    unfold Included; intros; apply H1 in H2; unfold Setminus in H2.
    apply Theorem4' in H2; apply H2.
Qed.

Theorem Theorem171 : ∀ (x: Class),
  Finite x -> Finite pow(x).
Proof.
  intros; double H.
  unfold Finite in H; apply Property_Finite in H0.
  assert (\{λ u, u ∈ ω /\ (∀ y, P[y] = u /\ Ensemble y -> Finite
  pow(y))\} = ω).
  { apply Theorem137.
    - unfold Included; intros; apply AxiomII in H1; apply H1.
    - apply AxiomII; generalize (Theorem135 x); intros; destruct H1.
      clear H2; repeat split; Ens; intros;
      destruct H2; AssE y; clear H3.
      generalize (classic (y = ∅)); intros; destruct H3.
      + assert (pow(∅) = [∅]).
        { unfold PowerSet, Singleton; apply AxiomI.
          split; intros; apply AxiomII in H5; destruct H5; apply AxiomII.
          - split; auto; intros; add (∅ ⊂ z) H6; try apply Theorem26.
            apply Theorem27 in H6; auto.
          - split; auto; rewrite H6; try apply Theorem19; Ens.
            unfold Included; intros; auto. }
        rewrite H3 in *; rewrite H5; apply Finite_Single; Ens.
      + apply Property_NotEmpty in H3; destruct H3;
        apply Theorem153 in H4.
        apply Theorem146 in H4; unfold Equivalent in H4.
        destruct H4 as [f H4], H4, H4, H5; rewrite <- H5 in H3.
        apply Property_Value in H3; auto; apply Property_ran in H3.
        rewrite H7, H2 in H3; generalize (Theorem16 f[x0]);
        contradiction.
    - intros; apply AxiomII in H1; apply AxiomII; destruct H1, H2;
      double H2.
      apply Theorem134 in H4; repeat split; Ens; intros; destruct H5.
      AssE y; clear H6; double H7; unfold PlusOne in H5;
      apply Theorem153 in H7.
      unfold Equivalent in H7; destruct H7 as [f H7], H7, H7, H8.
      assert (u ∈ P[y]).
      { rewrite H5; apply Theorem4; right; unfold Singleton.
        apply AxiomII; split; Ens. }
      double H11; rewrite <- H8 in H12;
```

```
apply Property_Value in H12; auto.
apply Property_ran in H12; rewrite H10 in H12;
apply Lemma171 in H12.
rewrite H12; clear H12; apply Theorem168.
assert (Finite pow(y ~ [f[u]])).
{ apply H3; assert (Ensemble (y ~ [f[u]])).
  { apply (Theorem33 y _); auto; unfold Included; intros.
    unfold Setminus in H12; apply Theorem4' in H12; apply H12. }
  repeat split; auto; intros; apply Theorem164 in H2.
  apply Theorem156 in H2; clear H1; destruct H2.
  rewrite <- H2 at 2; add (Ensemble u) H12;
  apply Theorem154 in H12.
  apply H12; clear H12; apply Theorem146; unfold Equivalent.
  exists (f|(P[y]~[u])).
  { repeat split; unfold Relation; intros.
    - unfold Restriction in H12; apply Theorem4' in H12.
      destruct H12; PP H13 a b; Ens.
    - unfold Restriction in H12; destruct H12;
      apply Theorem4' in H12.
      apply Theorem4' in H13; destruct H12, H13;
      unfold Function in H7.
      apply H7 with (x:= x0); split; auto.
    - PP H12 a b; Ens.
    - unfold Inverse, Restriction in H12; destruct H12.
      apply AxiomII_P in H12; apply AxiomII_P in H13;
      destruct H12, H13.
      apply Theorem4' in H14; apply Theorem4' in H15;
      destruct H14, H15.
      clear H16 H17;unfold Function in H9;apply H9 with (x:= x0).
      split; apply AxiomII_P; Ens.
    - apply AxiomI; split; intros.
      + unfold Domain in H12; apply AxiomII in H12;
        destruct H12, H13.
        unfold Restriction in H13; apply Theorem4' in H13;
        destruct H13.
        clear H13;unfold Cartesian in H14;apply AxiomII_P in H14.
        destruct H14, H14; clear H15; unfold Setminus in H14.
        apply Theorem4' in H14; destruct H14; rewrite H5 in H14.
        apply Theorem4 in H14; destruct H14; auto.
        apply AxiomII in H15; destruct H15; contradiction.
      + unfold Domain; apply AxiomII; split; Ens; exists f[z].
        unfold Restriction; apply Theorem4'.
        assert (z ∈ dom(f)).
        { rewrite H8, H5; apply Theorem4; tauto. }
        apply Property_Value in H13; auto; split; auto.
```

```
        unfold Cartesian; apply AxiomII_P; split; Ens; double H13.
        apply Property_ran in H14; split; try apply Theorem19; Ens.
        clear H14; apply Property_dom in H13; unfold Setminus.
        apply Theorem4'; rewrite H8 in H13; split; auto.
        unfold Complement; apply AxiomII; split; Ens; intro.
        apply AxiomII in H14; destruct H14.
        rewrite H15 in H12; try apply Theorem19; Ens.
        generalize (Theorem101 u); intros; contradiction.
    - apply AxiomI; split; intros.
      + unfold Range in H12; apply AxiomII in H12;
        destruct H12, H13.
        unfold Restriction in H13; apply Theorem4' in H13;
        destruct H13.
        unfold Cartesian in H14; apply AxiomII_P in H14;
        destruct H14.
        clear H14; destruct H15; clear H15; unfold Setminus in H14.
        apply Theorem4' in H14; destruct H14; double H14.
        rewrite <- H8 in H14; rewrite H5 in H16.
        apply Theorem4 in H16; destruct H16.
        * clear H16; apply Property_Value in H14; auto.
          add ([x0,z] ∈ f) H14; apply H7 in H14;
          rewrite <- H14 in *.
          clear H14; unfold Setminus; apply Theorem4'; double H13.
          apply Property_ran in H14; rewrite H10 in H14;
          split; auto.
          unfold Complement; apply AxiomII; split; Ens; intro.
          apply AxiomII in H16; destruct H16; assert (u ∈ dom(f)).
          { rewrite H8, H5; apply Theorem4; right.
            unfold Singleton; apply AxiomII; Ens. }
          apply Property_Value in H18; auto; AssE [u,f[u]].
          apply Theorem49 in H19; destruct H19.
          rewrite H17 in H13; try apply Theorem19; Ens.
          unfold Complement in H15; apply AxiomII in H15;
          destruct H15.
          elim H21; unfold Singleton; apply AxiomII;
          split; Ens; intros.
          apply H9 with (x:= f[u]); unfold Inverse.
          split; apply AxiomII_P; split; try apply Theorem49; auto.
        * unfold Complement in H15; apply AxiomII in H15.
          destruct H15; contradiction.
      + unfold Setminus in H12; apply Theorem4' in H12;
        destruct H12.
        rewrite <- H10 in H12; apply AxiomII in H12;
        destruct H12, H14.
        unfold Range; apply AxiomII; split; Ens; exists x0.
```

```
        unfold Restriction; apply Theorem4'; split; auto.
        unfold Cartesian; apply AxiomII_P; split; Ens.
        split; try apply Theorem19; Ens; apply Theorem4'.
        double H14; apply Property_dom in H15; rewrite <- H8;
        split; auto.
        unfold Complement; apply AxiomII; split; Ens; intro.
        apply AxiomII in H16; destruct H16.
        rewrite H17 in *; try apply Theorem19; Ens.
        apply Property_Value in H15; auto; clear H16 H17.
        add ([u,z] ∈ f) H15; apply H7 in H15; rewrite H15 in H13.
        unfold Complement in H13; apply AxiomII in H13; clear H12.
        destruct H13; elim H13; apply AxiomII; Ens. } }
split; auto.
assert (pow(y ~ [f[u]]) ≈ \{λ z, z ⊂ y /\ f[u] ∈ z\}).
{ unfold Equivalent.
  exists (\{\ λ v w, v ∈ pow(y~[f[u]]) /\ w = v ∪ [f[u]] \}\).
  repeat split; unfold Relation; intros; try PP H13 a b; Ens.
  - destruct H13; apply AxiomII_P in H13; apply AxiomII_P in H14.
    destruct H13, H14, H15, H16; rewrite H17, H18; auto.
  - destruct H13; apply AxiomII_P in H13; apply AxiomII_P in H14.
    destruct H13, H14; clear H13 H14; apply AxiomII_P in H15.
    apply AxiomII_P in H16; destruct H15, H16, H14, H16.
    rewrite H18 in H17; clear H13 H15 H18;
    unfold PowerSet in H14, H16.
    apply AxiomII in H14;apply AxiomII in H16;destruct H14,H16.
    apply AxiomI; split; intros.
    + assert (z0 ∈ (y0 ∪ [f[u]])). { apply Theorem4; tauto. }
      rewrite <- H17 in H19; apply Theorem4 in H19;
      destruct H19; auto.
      apply H14 in H18; unfold Setminus in H18;
      apply Theorem4' in H18.
      destruct H18;unfold Complement in H20;apply AxiomII in H20.
      destruct H20; contradiction.
    + assert (z0 ∈ (z ∪ [f[u]])). { apply Theorem4; tauto. }
      rewrite H17 in H19; apply Theorem4 in H19;destruct H19;auto.
      apply H16 in H18; unfold Setminus in H18;
      apply Theorem4' in H18.
      destruct H18; unfold Complement in H20;apply AxiomII in H20.
      destruct H20; contradiction.
  - unfold Domain; apply AxiomI; split; intros.
    + apply AxiomII in H13; destruct H13, H14.
      apply AxiomII_P in H14; apply H14.
    + apply AxiomII; split; Ens; exists (z ∪ [f[u]]).
      apply AxiomII_P; repeat split; auto; apply Theorem49.
      split; Ens; apply AxiomIV; split; Ens.
```

```
      rewrite <- H8 in H11; apply Property_Value in H11; auto.
      apply Property_ran in H11; AssE f[u]; apply Theorem42; auto.
  - unfold Range; apply AxiomI; split; intros.
    + apply AxiomII in H13; destruct H13, H14;
      apply AxiomII_P in H14.
      destruct H14, H15; apply AxiomII; split; auto; rewrite H16.
      clear H16; unfold PowerSet in H15; apply AxiomII in H15.
      destruct H15; rewrite <- H8 in H11.
      apply Property_Value in H11; auto; apply Property_ran in H11.
      rewrite H10 in H11; split.
      * unfold Included; intros; apply Theorem4 in H17;
        destruct H17.
        { apply H16 in H17; apply Theorem4' in H17; apply H17. }
        { apply AxiomII in H17; destruct H17.
          rewrite H18; try apply Theorem19; Ens. }
      * apply Theorem4; right; apply AxiomII; split; Ens.
    + apply AxiomII in H13; destruct H13, H14; apply AxiomII.
      split; auto; exists (z~[f[u]]); apply AxiomII_P.
      assert (Ensemble (z ~ [f[u]])).
      { apply Theorem33 with (x:= z); auto; unfold Included.
        intros; apply AxiomII in H16; apply H16. }
      repeat split.
      * apply Theorem49; split; auto.
      * unfold PowerSet; apply AxiomII; split; auto;
        unfold Setminus.
        unfold Included; intros; apply AxiomII in H17;
        destruct H17, H18.
        apply Theorem4'; split; auto.
      * apply AxiomI; split; intros.
        { generalize (classic (z0=f[u])); intros.
          apply Theorem4; destruct H18.
          - right; apply AxiomII; split; Ens.
          - left; unfold Setminus; apply Theorem4'; split; auto.
            apply AxiomII; split; Ens; intro; apply AxiomII in H19.
            destruct H19; rewrite H20 in H18; try tauto.
            apply Theorem19; Ens. }
        { apply Theorem4 in H17; destruct H17.
          - unfold Setminus in H17; apply Theorem4' in H17;
            apply H17.
          - unfold Singleton; apply AxiomII in H17; destruct H17.
            rewrite H18; auto; apply Theorem19; Ens. } }
double H12; apply Property_Finite in H14; unfold Finite in H12.
unfold Finite; apply Theorem154 in H13; try rewrite <- H13; auto.
split; auto; clear H13 H15; apply Theorem38 in H6; auto.
apply Theorem33 with (x:= pow(y)); auto;
```

```
    unfold Included at 1; intros.
    apply AxiomII in H13; destruct H13, H14, H15; unfold PowerSet.
    apply AxiomII; split; auto. }
  rewrite <- H1 in H; clear H1; apply AxiomII in H; destruct H, H1.
  apply H2; split; auto.
Qed.

Hint Resolve Theorem171 : set.
```

定理 172 x 有限 $\wedge\, y \subset x\ \wedge P(y) = P(x) \implies x = y$.

```
Theorem Theorem172 : ∀ x y,
  Finite x -> y ⊂ x -> P[y] = P[x] -> x = y.
Proof.
  intros.
  double H; apply Property_Finite in H2; symmetry.
  double H; unfold Finite in H3; unfold ω in H3; apply AxiomII in H3.
  destruct H3; unfold Integer in H4; destruct H4;
  unfold WellOrdered in H5.
  double H2; apply Theorem153 in H6; apply Theorem146 in H6.
  unfold Equivalent in H6; destruct H6 as [f H6], H6, H6, H7.
  generalize (classic (y = x)); intros; destruct H10; auto.
  assert (y ⊂ x). { unfold ProperIncluded; split; auto. }
  apply Property_ProperIncluded' in H11; destruct H11 as [z H11], H11.
  generalize (classic (P[x] = ∅)); intros; destruct H13.
  - rewrite <- H7 in H11; apply Property_Value in H11; auto.
    apply Property_ran in H11; rewrite H9, H13 in H11.
    generalize (Theorem16 f[z]); intros; contradiction.
  - assert (P[x] ⊂ P [x] /\ P[x] ≠ ∅).
  { split; unfold Included; Ens. }
    apply H5 in H14; clear H5 H13; destruct H14 as [u H5].
    assert (P[x] ∈ R /\ LastMember u E P[x]).
    { split; auto; unfold R; apply AxiomII; split; auto. }
    apply Theorem133 in H13; unfold FirstMember in H5;
    destruct H5; clear H14.
    assert ((x ~ [z]) ≈ u).
    { rewrite <- H9 in H5; apply AxiomII in H5; destruct H5; clear H5.
      destruct H14; rewrite <- H7 in H11;
      apply Property_Value in H11; auto.
      generalize (classic (z = x0)); intros; destruct H14.
      - rewrite H14; unfold Equivalent; exists (f | (x ~ [x0])).
        repeat split; unfold Relation; intros.
        + apply Theorem4' in H15; destruct H15; PP H16 a b; Ens.
        + unfold Restriction in H15; destruct H15;
          apply Theorem4' in H15.
```

```
  apply Theorem4' in H16; destruct H15, H16;
  apply H6 with (x:= x1); auto.
+ PP H15 a b; Ens.
+ unfold Inverse, Restriction in H15; destruct H15.
  apply AxiomII_P in H15; apply AxiomII_P in H16;
  destruct H15, H16.
  apply Theorem4' in H17; apply Theorem4' in H18;
  destruct H17, H18.
  apply H8 with (x:= x1); unfold Inverse; split;
  apply AxiomII_P; Ens.
+ apply AxiomI; split; intros.
  * unfold Domain in H15; apply AxiomII in H15;
    destruct H15, H16.
    unfold Restriction in H16; apply Theorem4' in H16;
    destruct H16.
    unfold Cartesian in H17; apply AxiomII_P in H17; apply H17.
  * unfold Domain; apply AxiomII; split; Ens; exists f[z0];
    double H15.
    unfold Setminus in H16; apply Theorem4' in H16; destruct H16.
    clear H17; rewrite <- H7 in H16;
    apply Property_Value in H16; auto.
    unfold Restriction; apply Theorem4';
    split; auto; unfold Cartesian.
    apply AxiomII_P; repeat split; Ens;
    apply Property_ran in H16.
    apply Theorem19; Ens.
+ apply AxiomI; split; intros.
  * unfold Range in H15; apply AxiomII in H15; destruct H15,H16.
    apply Theorem4' in H16; destruct H16; double H16.
    apply Property_ran in H18; rewrite H9, H13 in H18.
    unfold PlusOne in H18; apply Theorem4 in H18;
    destruct H18; auto.
    apply AxiomII in H18; destruct H18; unfold Cartesian in H17.
    apply AxiomII_P in H17; destruct H17, H20; clear H17 H18 H21.
    unfold Setminus in H20; apply Theorem4' in H20; destruct H20.
    clear H17; unfold Complement in H18; apply AxiomII in H18.
    destruct H18; elim H18; clear H18;
    unfold Singleton; apply AxiomII.
    split; auto; intros; clear H17 H18; apply H8 with (x:= u).
    AssE [x0,u]; apply Theorem49 in H17; destruct H17.
    rewrite H19 in H16; try apply Theorem19; Ens; clear H19.
    split; apply AxiomII_P; split; try apply Theorem49; auto.
    apply Property_dom in H16; split; Ens.
  * unfold Range; apply AxiomII; split; Ens;
    assert (z0 ∈ ran(f)).
```

```
      { rewrite H9, H13; unfold PlusOne; apply Theorem4; tauto. }
      unfold Range in H16;apply AxiomII in H16;destruct H16,H17.
      exists x1; unfold Restriction; apply Theorem4'; split; auto.
      unfold Cartesian; apply AxiomII_P; split; Ens; double H17.
      split; try apply Theorem19; auto; unfold Setminus.
      apply Property_dom in H18;rewrite H7 in H18;apply Theorem4'.
      split; auto; unfold Complement; apply AxiomII;
      split; Ens; intro.
      unfold Singleton in H19; apply AxiomII in H19; destruct H19.
      double H5; apply Property_dom in H21; clear H18 H19.
      rewrite H20 in H17; try apply Theorem19; Ens; clear H20 H21.
      add ([x0,z0] ∈ f) H5; apply H6 in H5; clear H17.
      rewrite H5 in H15;generalize (Theorem101 z0);contradiction.
  - unfold Equivalent; exists (\{\ λ v w, v ∈ (x ~ [z]) /\
    (v = x0 -> w = f[z]) /\ (v <> x0 -> [v,w] ∈ f) \}\).
    repeat split; unfold Relation; intros; try PP H15 a b; Ens.
    + destruct H15; apply AxiomII_P in H15; apply AxiomII_P in H16.
      destruct H15, H16, H17, H18;
      generalize (classic (x1 = x0)); intros.
      destruct H21; double H21; apply H19 in H21; apply H20 in H22.
      * rewrite H21, H22; auto.
      * apply H6 with (x:= x1); auto.
    + destruct H15; apply AxiomII_P in H15; apply AxiomII_P in H16.
      destruct H15, H16; apply AxiomII_P in H17;
      apply AxiomII_P in H18.
      destruct H17, H18; clear H17 H18; destruct H19, H20.
      apply Theorem4' in H17; apply Theorem4' in H19;
      destruct H17, H19.
      generalize (classic (y0 = x0)) (classic (z0 = x0)); intros.
      destruct H23, H24; try rewrite H23, H24; apply H18 in H23;
      apply H20 in H24; clear H18 H20; auto.
      * rewrite <- H23 in H11; clear H23.
        assert ([x1,z] ∈ f⁻¹ /\ [x1,z0] ∈ f⁻¹).
        { unfold Inverse; split; apply AxiomII_P; split; Ens.
          AssE [z,x1]; apply Theorem49; apply Theorem49 in H18;
          tauto. }
        apply H8 in H18; rewrite <- H18 in H22; clear H18 H24.
        apply AxiomII in H22; destruct H22; elim H20;
        apply AxiomII; Ens.
      * rewrite <- H24 in H11; clear H24.
        assert ([x1,z] ∈ f⁻¹ /\ [x1,y0] ∈ f⁻¹).
        { unfold Inverse; split; apply AxiomII_P; split; Ens.
          AssE [z,x1]; apply Theorem49; apply Theorem49 in H18;
          tauto. }
        apply H8 in H18; rewrite <- H18 in H21;
```

```
    clear H16 H18 H19 H22 H23.
    apply AxiomII in H21; destruct H21; elim H18;
    apply AxiomII; Ens.
  * apply H8 with (x:= x1); split; apply AxiomII_P; split; auto.
+ apply AxiomI; split; intros.
  * unfold Domain in H15; apply AxiomII in H15;
    destruct H15, H16.
    apply AxiomII_P in H16; apply H16.
  * unfold Domain; apply AxiomII; split; Ens.
    generalize (classic (z0 = x0)); intros; destruct H16.
    { exists f[z]; apply AxiomII_P; repeat split;
      intros; try tauto.
      apply Property_ran in H11; apply Theorem49; split; Ens. }
    { exists f[z0]; double H15; apply Theorem4' in H17;
      destruct H17.
      clear H18; rewrite <- H7 in H17;
      apply Property_Value in H17; auto.
      apply AxiomII_P; repeat split; intros; try tauto; Ens. }
+ apply AxiomI; split; intros.
  * unfold Range in H15;apply AxiomII in H15;destruct H15, H16.
    apply AxiomII_P in H16; destruct H16, H17.
    generalize (classic (x1 = x0)); intros; destruct H19.
    { apply H18 in H19; clear H18; rewrite H19; double H11.
      apply Property_ran in H18; rewrite H9, H13 in H18.
      apply Theorem4 in H18; destruct H18; auto; AssE [x0, u].
      apply Theorem49 in H20;destruct H20;apply AxiomII in H18.
      destruct H18;rewrite H22 in H11;try apply Theorem19;auto.
      elim H14; apply H8 with (x:= u); unfold Inverse.
      split; apply AxiomII_P; split; try apply Theorem49; Ens.
      apply Property_dom in H11; split; Ens. }
    { double H19; apply H18 in H20; clear H18; double H20.
      apply Property_ran in H20; rewrite H9, H13 in H20;
      clear H15.
      apply Theorem4 in H20; destruct H20; auto;
      apply AxiomII in H15.
      destruct H15; AssE [x0,u]; apply Theorem49 in H21;
      destruct H21.
      rewrite H20 in H18; try apply Theorem19; Ens; clear H20.
      elim H19; apply H8 with (x:= u); unfold Inverse.
      split; apply AxiomII_P; split; try apply Theorem49; Ens. }
  * unfold Range; apply AxiomII; split; Ens.
    assert (z0 ∈ ran(f)).
    { rewrite H9, H13; apply Theorem4; tauto. }
    unfold Range in H16; apply AxiomII in H16; destruct H16,H17.
    generalize (classic (z0 = f[z])); intros; destruct H18.
```

```
        { exists x0; apply Property_dom in H5; apply AxiomII_P.
          repeat split; intros; try tauto; try apply Theorem49; Ens.
          rewrite H7 in H5; unfold Setminus;
          apply Theorem4'; split; auto.
          unfold Complement; apply AxiomII; split; Ens; intro.
          unfold Singleton in H19; apply AxiomII in H19;
          destruct H19.
          apply Property_dom in H11; rewrite H20 in H14; try tauto.
          apply Theorem19; Ens. }
        { exists x1; apply AxiomII_P; repeat split; intros; Ens.
          - double H17; apply Property_dom in H19; rewrite H7 in H19.
            unfold Setminus; apply Theorem4'; split; auto.
            apply AxiomII; split; Ens; intro; apply AxiomII in H20.
            destruct H20; elim H18; apply H6 with (x:= z).
            rewrite H21 in H17; try split; auto; apply Theorem19.
            apply Property_dom in H11; Ens.
          - rewrite H19 in H17; add ([x0,z0] ∈ f) H5; apply H6 in H5.
            rewrite H5 in H15; generalize (Theorem101 z0);
            contradiction. }}
    assert (Ensemble (x ∼ [z]) /\ y ⊂ (x ∼ [z])).
    { split.
      - apply (Theorem33 x _); auto; unfold Included; intros.
        unfold Setminus in H15; apply Theorem4' in H15; apply H15.
      - unfold Included, Setminus; intros; apply Theorem4'; split; auto.
        unfold Complement; apply AxiomII; split; Ens;
        unfold Singleton; intro.
        apply AxiomII in H16; destruct H16; rewrite H17 in H15;
        try contradiction.
        apply Theorem19; Ens. }
    elim H15; intros; apply Theorem158 in H15; rewrite H1, H13 in H15.
    clear H17; add (Ensemble u) H16; Ens; apply Theorem154 in H16.
    apply H16 in H14; rewrite H14 in H15; clear H3 H13 H14 H16.
    unfold Finite in H; unfold ω in H; apply AxiomII in H; destruct H.
    AssE u; apply Theorem132 in H5; auto.
    assert (u ∈ C). { apply Theorem164; apply AxiomII; split; auto. }
    clear H5 H13; apply Theorem156 in H14; destruct H14;
    rewrite H13 in H15.
    assert (u ∈ (u ∪ [u])).
    { apply Theorem4; right; apply AxiomII; Ens. }
    clear H5 H13; unfold PlusOne, LessEqual in H15; destruct H15.
    + generalize (Theorem102 (u ∪ [u]) u); intros; destruct H13; auto.
    + rewrite H5 in H14; generalize (Theorem101 u); contradiction.
Qed.

Hint Resolve Theorem172 : set.
```

定理 172 是关于有限集不能等势于它的真子集的性质, 实际上, 它描述了有限集的特征.

定理 173　x 是集且非有限 $\Longrightarrow$ $(\exists\, y,\ y \subset x,\ y \neq x,\ x \approx y)$.

```
Lemma Lemma173 : ∀ x0 x,
  x0 ∈ P [x] -> ∼ x0 ∈ ω -> x0 ∈ (P [x] ∼ ω).
Proof.
  intros; unfold Setminus; apply Theorem4'; split; auto.
  unfold Complement; apply AxiomII; split; Ens.
Qed.

Theorem Theorem173 : ∀ x,
  Ensemble x /\ ∼ Finite x -> (exists y, y ⊂ x /\ y ∉ x /\ x ≈ y).
Proof.
  intros; destruct H.
  assert (ω ⊂ P[x]).
  { unfold Included; intros; double H; apply Property_PClass in H2.
    unfold ω in H1; unfold C in H2; apply AxiomII in H1;
    apply AxiomII in H2.
    destruct H1, H2, H4; clear H5;
    unfold Ordinal_Number in H4; double H3.
    unfold Integer in H5; destruct H5; apply AxiomII in H4; clear H2 H6.
    destruct H4; add (Ordinal z) H4; clear H5; apply Theorem110 in H4.
    destruct H4 as [H4 | [H4 | H4]]; auto.
    - apply Theorem132 in H4; auto; destruct H0; unfold Finite.
      unfold ω; apply AxiomII; split; auto.
    - destruct H0; unfold Finite; rewrite H4; apply AxiomII; Ens. }
  assert (P[x] ≈ (P[x] ∼ [∅])).
  { unfold Equivalent; exists (\{\ λ u v, u ∈ P[x] /\ ((u ∈ ω -> v =
    PlusOne u) /\ (u ∈ (P[x] ∼ ω) -> v = u)) \}\).
    repeat split; unfold Relation; intros; try PP H2 a b; Ens.
    - destruct H2; apply AxiomII_P in H2; apply AxiomII_P in H3.
      destruct H2, H3, H4, H5;
      generalize (classic (x0 ∈ ω)); intros; destruct H8.
      + double H8; apply H6 in H8; apply H7 in H9; rewrite H8, H9; auto.
      + apply Lemma173 in H5; auto; clear H8; double H5.
        apply H6 in H5; apply H7 in H8; rewrite H5, H8; auto.
    - destruct H2; apply AxiomII_P in H2; apply AxiomII_P in H3;
      destruct H2, H3.
      apply AxiomII_P in H4; apply AxiomII_P in H5;
      destruct H4, H5, H6, H7.
      generalize (classic (y ∈ ω)) (classic (z ∈ ω)); intros;
      destruct H10, H11.
      + apply Theorem136; auto; apply H8 in H10; apply H9 in H11.
```

```
      rewrite H10 in H11; auto.
    + double H10; apply Lemma173 in H7; auto; apply H8 in H10;
      apply H9 in H7.
      rewrite H7 in H10; rewrite H10 in H11;
      apply Theorem134 in H12; tauto.
    + double H11; apply Lemma173 in H6; auto; apply H8 in H6;
      apply H9 in H11.
      rewrite H6 in H11; rewrite H11 in H10;
      apply Theorem134 in H12; tauto.
    + apply Lemma173 in H6; apply Lemma173 in H7; auto.
      apply H8 in H6; apply H9 in H7; rewrite <- H6, <- H7; auto.
  - apply AxiomI; split; intros.
    + unfold Domain in H2; apply AxiomII in H2; destruct H2, H3.
      apply AxiomII_P in H3; apply H3.
    + unfold Domain; apply AxiomII; split; Ens.
      generalize (classic (z ∈ ω)); intros; destruct H3.
      * exists (PlusOne z); apply AxiomII_P;
        repeat split; intros; auto.
        { apply Theorem134 in H3; apply Theorem49; split; Ens. }
        { unfold Setminus in H4; apply Theorem4' in H4;destruct H4.
          apply AxiomII in H5; destruct H5; contradiction. }
      * exists z; apply AxiomII_P; repeat split;
        try apply Theorem49; Ens.
        intros; contradiction.
  - apply AxiomI; split; intros.
    + unfold Range in H2; apply AxiomII in H2; destruct H2, H3.
      apply AxiomII_P in H3; destruct H3, H4.
      generalize (classic (x0 ∈ ω)); intros; destruct H6.
      * double H6; apply H5 in H7; clear H5; double H6; double H6.
        apply Theorem134 in H6; apply Theorem135 in H8;
        rewrite H7 in *.
        clear H7; apply H1 in H6; unfold Setminus; apply Theorem4'.
        split; auto; unfold Complement; apply AxiomII; split; Ens.
        intro; apply AxiomII in H7;destruct H7;rewrite H9 in H8;auto.
        apply Theorem19; Ens; exists ω; apply Theorem135; auto.
      * double H4; apply Lemma173 in H7; auto;
        apply H5 in H7; clear H5.
        rewrite H7 in *; clear H7; unfold Setminus; apply Theorem4'.
        split; auto; unfold Complement; apply AxiomII; split; Ens.
        intro; apply AxiomII in H5; clear H2; destruct H5.
        generalize (Theorem135 x0); intros; destruct H7; clear H8.
        rewrite H5 in H6; try apply Theorem19; Ens.
    + unfold Setminus in H2; apply Theorem4' in H2; destruct H2.
      unfold Complement in H3; apply AxiomII in H3; destruct H3.
      unfold Range; apply AxiomII; split; Ens.
```

```
      generalize (classic (z ∈ ω)); intros; destruct H5.
      * unfold ω in H5; apply AxiomII in H5; destruct H5; double H6.
        unfold Integer in H7; destruct H7, H8; clear H8.
        assert (z ⊂ z /\ z ∉ ∅).
        { split; unfold Included; intros; auto; intro; destruct H4.
          generalize (Theorem135 z); intros; destruct H4; rewrite H8.
          clear H8 H10; apply AxiomII; split; Ens. }
        apply H9 in H8; clear H9; destruct H8.
        assert (z ∈ R /\ LastMember x0 E z).
        { split; auto; try apply AxiomII; Ens. }
        apply Theorem133 in H9;unfold FirstMember in H8;destruct H8.
        AssE x0; apply Theorem132 in H8; auto; clear H10; exists x0.
        apply AxiomII_P; repeat split; intros;
        try apply Theorem49; Ens.
        -- apply H1; apply AxiomII; Ens.
        -- unfold Setminus in H10; apply Theorem4' in H10;
           destruct H10.
           unfold Complement in H12; apply AxiomII in H12;
           destruct H12, H13.
           unfold ω; apply AxiomII; split; auto.
      * exists z; apply AxiomII_P; repeat split;
        try apply Theorem49; Ens.
        intros; contradiction. }
double H; apply Theorem153 in H3; unfold Equivalent in H3.
destruct H3 as [f H3], H3, H3, H4.
assert (P[x] ~ [∅] ≈ ran(f | (P[x] ~ [∅]))).
{ unfold Equivalent; exists (f | (P[x] ~ [∅])).
  repeat split; unfold Relation; intros; try PP H7 a b; Ens.
  - unfold Restriction in H7; apply Theorem4' in H7; destruct H7.
    PP H8 a b; Ens.
  - unfold Restriction in H7; destruct H7; apply Theorem4' in H7.
    apply Theorem4' in H8; destruct H7, H8;
    apply H3 with (x:= x0); auto.
  - unfold Restriction, Inverse in H7; destruct H7;
    apply AxiomII_P in H7.
    apply AxiomII_P in H8; destruct H7, H8; apply Theorem4' in H9.
    apply Theorem4' in H10; destruct H9, H10; apply H5 with (x:= x0).
    unfold Inverse; split; apply AxiomII_P; split; Ens.
  - apply AxiomI; split; intros.
    + unfold Domain in H7; apply AxiomII in H7; destruct H7, H8.
      unfold Restriction in H8; apply Theorem4' in H8; destruct H8.
      unfold Cartesian in H9; apply AxiomII_P in H9; apply H9.
    + unfold Domain; apply AxiomII; split; Ens; exists f[z].
      unfold Restriction; apply Theorem4'; double H7;
      unfold Setminus in H8.
```

```
      apply Theorem4' in H8; destruct H8; clear H9;
      rewrite <- H4 in H8.
      apply Property_Value in H8; auto; split; auto; unfold Cartesian.
      apply AxiomII_P; AssE ([z,f[z]]); apply Property_ran in H8.
      repeat split; Ens; apply Theorem19; Ens. }
  double H; apply Theorem153 in H8; apply Theorem146 in H8.
  apply Theorem147 with (z:= P[x] ~ [∅]) in H8; auto; clear H2.
  apply Theorem147 with (z:= ran(f | (P[x] ~ [∅]))) in H8; auto;
  clear H7.
  exists ran(f | (P[x] ~ [∅])); repeat split; auto.
  - unfold Included; intros; unfold Range, Restriction in H2.
    apply AxiomII in H2; destruct H2, H7; apply Theorem4' in H7;
    destruct H7.
    apply Property_ran in H7; rewrite H6 in H7; auto.
  - generalize (Theorem135 x); intros; destruct H2; clear H7;
    apply H1 in H2.
    rewrite <- H4 in H2; apply Property_Value in H2; auto; intro.
    assert (f[∅] ∈ ran(f | (P[x] ~ [∅]))).
    { rewrite H7; apply Property_ran in H2; rewrite H6 in H2; auto. }
    unfold Range in H9; apply AxiomII in H9; destruct H9, H10.
    unfold Restriction in H10; apply Theorem4' in H10; destruct H10.
    assert ([f[∅],∅] ∈ f⁻¹ /\ [f[∅],x0] ∈ f⁻¹).
    { AssE [∅,f[∅]]; AssE [x0,f[∅]]; apply Theorem49 in H12.
      apply Theorem49 in H13; destruct H12, H13; clear H14 H15.
      split; apply AxiomII_P; split; try apply Theorem49; auto. }
    apply H5 in H12; rewrite H12 in H11; clear H12;
    unfold Cartesian in H11.
    apply AxiomII_P in H11; destruct H11, H12; clear H11 H13.
    unfold Setminus in H12; apply Theorem4' in H12; destruct H12.
    unfold Complement in H12; apply AxiomII in H12; destruct H12, H13.
    unfold Singleton; apply AxiomII; split; Ens.
Qed.

Hint Resolve Theorem173 : set.
```

定理 174 $x \in R \sim \omega \implies P(x+1) = P(x).$

```
Lemma Lemma174 : ∀ x y, x ≼ y -> y ≼ x -> x = y.
Proof.
  intros; unfold LessEqual in H, H0; destruct H, H0; auto.
  generalize (Theorem102 x y); intros; destruct H1; auto.
Qed.

Theorem Theorem174 : ∀ x,
  x ∈ (R ~ ω) -> P[ PlusOne x ] = P[x].
```

```
Proof.
  intros.
  unfold Setminus in H; apply Theorem4' in H; destruct H.
  double H; apply Lemma123 in H1; AssE (PlusOne x).
  add (x ⊂ (PlusOne x)) H2;
  try (unfold Included; intros; apply Theorem4; tauto).
  apply Theorem158 in H2; apply AxiomII in H0; destruct H0.
  assert (Ensemble x /\ ~ Finite x).
  { split; auto; clear H1 H2; unfold Finite; intro; destruct H3.
    generalize (Property_ω); intros; unfold R in H; apply AxiomII in H.
    clear H0; destruct H; assert (Ordinal ω /\ Ordinal x); auto;
    double H3.
    apply Theorem110 in H3; apply Theorem118 in H4.
    destruct H3 as [H3 | [H3 | H3]]; auto.
    - assert (ω ≼ x); unfold LessEqual; try tauto; apply H4 in H5;
      clear H4.
      assert (Ensemble x /\ ω ⊂ x); auto; apply Theorem158 in H4.
      double H1; unfold ω in H6; apply AxiomII in H6; destruct H6.
      unfold Integer in H7; destruct H7; clear H8.
      assert (Ordinal P[x] /\ Ordinal ω); auto; apply Theorem118 in H8.
      assert (P[x] ≼ ω); unfold LessEqual; try auto; apply H8 in H9;
      clear H8.
      assert (Ensemble ω /\ P [x] ⊂ ω); Ens; apply Theorem158 in H8;
      clear H9.
      rewrite Theorem155 in H8. apply Lemma174 in H8; auto.
      rewrite <- H8 in H1; clear H4 H8; generalize Theorem165; intros.
      apply Theorem156 in H4; destruct H4; rewrite H8 in H1.
      generalize (Theorem101 ω); intros; contradiction.
    - rewrite H3 in H1; generalize Theorem165; intros; rewrite H3 in H5.
      apply Theorem156 in H5; destruct H5; rewrite H6 in H1.
      generalize (Theorem101 x); intros; contradiction. }
  apply Theorem173 in H4; destruct H4 as [u H4], H4, H5.
  assert (P[PlusOne x] ≼ P[x]).
  { assert (u ⊊ x). { split; auto. }
  apply Property_ProperIncluded' in H7.
    destruct H7 as [z H7], H7, H6 as [f H6], H6, H6, H9.
    assert (PlusOne x ≈ ran(\{\ λ v w, (v∈x /\ w=f[v]) \/ (v=x /\ w=z)
    \}\)).
    { unfold Equivalent;
      exists (\{\ λ v w, (v∈x /\ w=f[v]) \/ (v=x /\ w=z)\}\).
      repeat split; unfold Relation; intros; try PP H12 a b; Ens.
      - destruct H12; apply AxiomII_P in H12; apply AxiomII_P in H13.
        destruct H12, H13, H14, H15, H14, H15;
        try rewrite H16, H17; auto.
        + rewrite H15 in H14; generalize (Theorem101 x); contradiction.
```

```
    + rewrite H14 in H15; generalize (Theorem101 x); contradiction.
  - destruct H12; apply AxiomII_P in H12; apply AxiomII_P in H13.
    destruct H12, H13; apply AxiomII_P in H14;
    apply AxiomII_P in H15.
    destruct H14, H15, H16, H17, H16, H17.
    + rewrite <- H9 in H16, H17; apply Property_Value in H16; auto.
      apply Property_Value in H17; auto; rewrite H19 in *; clear H19.
      rewrite H18 in *; clear H18; apply H10 with (x:= f[y]).
      unfold Inverse; split; apply AxiomII_P; Ens.
    + rewrite <- H9 in H16; apply Property_Value in H16; auto.
      apply Property_ran in H16; rewrite H11 in H16;
      rewrite <- H18 in H16.
      rewrite H19 in H16; contradiction.
    + rewrite <- H9 in H17; apply Property_Value in H17; auto.
      apply Property_ran in H17; rewrite H11 in H17;
      rewrite <- H19 in H17.
      rewrite H18 in H17; contradiction.
    + rewrite H16, H17; auto.
  - apply AxiomI; split; intros.
    + unfold PlusOne; apply Theorem4; unfold Domain in H12.
      apply AxiomII in H12; destruct H12, H13;
      apply AxiomII_P in H13.
      destruct H13, H14; destruct H14; try tauto; rewrite H14 in *.
      right; unfold Singleton; apply AxiomII; split; Ens.
    + unfold Domain; apply AxiomII; split; Ens;
      unfold PlusOne in H12.
      apply Theorem4 in H12; destruct H12.
      * double H12; rewrite <- H9 in H13;
        apply Property_Value in H13; auto.
        exists f[z0]; apply AxiomII_P; repeat split; intros; Ens.
      * exists z; apply AxiomII_P; split; try apply Theorem49; Ens.
        right; split; auto; unfold Singleton in H12;
        apply AxiomII in H12.
        apply H12; apply Theorem19; auto. }
assert (Ensemble x /\ ran(\{\ λ v w, (v∈x/\w=f[v]) \/
(v=x/\w=z)\}\) ⊂ x).
{ split; auto; unfold Included; intros; apply AxiomII in H13.
  destruct H13, H14; apply AxiomII_P in H14; destruct H14, H15, H15.
  - rewrite H16; rewrite <- H9 in H15;
    apply Property_Value in H15; auto.
    apply Property_ran in H15; rewrite H11 in H15; auto.
  - rewrite H16; auto. }
clear H0; elim H13; intros; apply Theorem158 in H13.
apply Theorem154 in H12;
try (split; Ens; apply (Theorem33 x _); auto).
```

```
    rewrite <- H12 in H13; clear H12 H14; auto. }
  apply Lemma174; auto.
Qed.

Hint Resolve Theorem174 : set.
```

定义 175　$\max[x, y] = x \cup y$.

```
Definition Max x y : Class := x ∪ y.

Corollary Property_Max : ∀ x y,
  (Ordinal x -> Ordinal y -> x ∈ y -> Max x y = y) /\
  (x = y -> Max x y = y).
Proof.
  split; intros; try (rewrite H; unfold Max; apply Theorem5).
  assert (x ≼ y); unfold LessEqual; try tauto.
  apply Theorem118 in H2; auto; unfold Max; apply Theorem29; auto.
Qed.

Corollary Equal_Max : ∀ x y, Max x y = Max y x.
Proof.
  intros; unfold Max; apply Theorem6.
Qed.

Hint Unfold Max : set.
```

定义 176　$\ll = \{z : \exists (u, v) \in R \times R, \exists (x, y) \in R \times R, z = ((u, v), (x, y)), (\max[u, v] < \max[x, y]) \vee (\max[u, v] = \max[x, y] \wedge u < x) \vee (\max[u, v] = \max[x, y] \wedge u = x \wedge v < y)\}$.

```
Definition LessLess : Class :=
  \{ λ z, ∃ u v x y, [u,v]∈(R×R) /\ [x,y]∈(R×R) /\ z = [[u,v],[x,y]]
  /\ ((Max u v ≺ Max x y) \/ (Max u v = Max x y /\ u ≺ x) \/ (Max u v
  =Max x y /\ u = x /\ v ≺ y)) \}.

Notation "≪" := (LessLess) (at level 0, no associativity).

Hint Unfold LessLess : set.
```

定理 177　$\ll$ 良序 $R \times R$.

```
Definition En_y y : Class := \{ λ z, exists u v,
  [u,v] ∈ y /\ z = Max u v \}.

Definition En_v v y : Class := \{ λ z, [z,v] ∈ y /\ z ∈ v \}.
```

```
Definition En_u u y : Class := \{ λ z, [u,z] ∈ y /\ z ∈ u \}.

Lemma Lemma177_bd : ∀ a b c d,
  [a, b] ∈ R × R -> [c, d] ∈ R × R -> Ensemble a -> Ensemble b ->
  Ensemble c -> Ensemble d -> Ordinal a -> Ordinal b -> Ordinal c ->
  Ordinal d -> Max a b = b -> Max c d = d ->
  Rrelation ([a,b]) ≪ ([c,d]) \/ Rrelation ([c,d]) ≪ ([a,b]) \/
  [a,b] = [c,d].
Proof.
  intros.
  assert (Ordinal b /\ Ordinal d); auto; apply Theorem110 in H11.
  destruct H11 as [H11 | [H11 | H11]].
  - left; unfold Rrelation, LessLess; apply AxiomII.
    split; try (apply Theorem49; split; apply Theorem49; auto).
    exists a, b, c, d; repeat split; auto; left.
    rewrite H9, H10; unfold Less; auto.
  - right; left; unfold Rrelation, LessLess; apply AxiomII.
    split; try (apply Theorem49; split; apply Theorem49; auto).
    exists c, d, a, b; repeat split; auto; left.
    rewrite H9, H10; unfold Less; auto.
  - assert (Ordinal a /\ Ordinal c); auto; apply Theorem110 in H12.
    destruct H12 as [H12 | [H12 | H12]].
    { left; unfold Rrelation, LessLess; apply AxiomII.
      split; try (apply Theorem49; split; apply Theorem49; auto).
      exists a, b, c, d; repeat split; auto; right; left.
      rewrite H9, H10; unfold Less; split; auto. }
    { right; left; unfold Rrelation, LessLess; apply AxiomII.
      split; try (apply Theorem49; split; apply Theorem49; auto).
      exists c, d, a, b; repeat split; auto; right; left.
      rewrite H9, H10; unfold Less; split; auto. }
    { right; right; rewrite H11, H12; auto. }
Qed.

Lemma Lemma177_bc : ∀ a b c d,
  [a, b] ∈ R × R -> [c, d] ∈ R × R -> Ensemble a -> Ensemble b ->
  Ensemble c -> Ensemble d -> Ordinal a -> Ordinal b -> Ordinal c ->
  Ordinal d -> Max a b = b -> Max c d = c ->
  Rrelation ([a,b]) ≪ ([c,d]) \/ Rrelation ([c,d]) ≪ ([a,b]) \/
  [a,b] = [c,d].
Proof.
  intros; assert (Ordinal b /\ Ordinal c); auto.
  apply Theorem110 in H11; destruct H11 as [H11 | [H11 | H11]].
  - left; unfold Rrelation, LessLess; apply AxiomII.
    split; try (apply Theorem49; split; apply Theorem49; auto).
```

```
    exists a, b, c, d; repeat split; auto; left.
    rewrite H9, H10; unfold Less; auto.
  - right; left; unfold Rrelation, LessLess; apply AxiomII.
    split; try (apply Theorem49; split; apply Theorem49; auto).
    exists c, d, a, b; repeat split; auto; left.
    rewrite H9, H10; unfold Less; auto.
  - assert (Ordinal a /\ Ordinal c); auto; apply Theorem110 in H12.
    destruct H12 as [H12 | [H12 | H12]].
    { left; unfold Rrelation, LessLess; apply AxiomII.
      split; try (apply Theorem49; split; apply Theorem49; auto).
      exists a, b, c, d; repeat split; auto; right; left.
      rewrite H9, H10; unfold Less; split; auto. }
    { right; left; unfold Rrelation, LessLess; apply AxiomII.
      split; try (apply Theorem49; split; apply Theorem49; auto).
      exists c, d, a, b; repeat split; auto; right; left.
      rewrite H9, H10; unfold Less; split; auto. }
    { assert (Ordinal b /\ Ordinal d); auto; apply Theorem110 in H13.
      destruct H13 as [H13 | [H13 | H13]].
      - left; unfold Rrelation, LessLess; apply AxiomII.
        split; try (apply Theorem49; split; apply Theorem49; auto).
        exists a, b, c, d; repeat split; auto; right; right.
        rewrite H9, H10; unfold Less; auto.
      - right; left; unfold Rrelation, LessLess; apply AxiomII.
        split; try (apply Theorem49; split; apply Theorem49; auto).
        exists c, d, a, b; repeat split; auto; right; right.
        rewrite H9, H10; unfold Less; auto.
      - right; right; rewrite H12, H13; auto. }
Qed.

Lemma Lemma177_ac : ∀ a b c d,
  [a, b] ∈ R × R -> [c, d] ∈ R × R -> Ensemble a -> Ensemble b ->
  Ensemble c -> Ensemble d -> Ordinal a -> Ordinal b -> Ordinal c ->
  Ordinal d -> Max a b = a -> Max c d = c ->
  Rrelation ([a,b]) ≪ ([c,d]) \/ Rrelation ([c,d]) ≪ ([a,b]) \/
  [a,b] = [c,d].
Proof.
  intros; assert (Ordinal a /\ Ordinal c); auto.
  apply Theorem110 in H11; destruct H11 as [H11 | [H11 | H11]].
  - left; unfold Rrelation, LessLess; apply AxiomII.
    split; try (apply Theorem49; split; apply Theorem49; auto).
    exists a, b, c, d; repeat split; auto; left.
    rewrite H9, H10; unfold Less; auto.
  - right; left; unfold Rrelation, LessLess; apply AxiomII.
    split; try (apply Theorem49; split; apply Theorem49; auto).
    exists c, d, a, b; repeat split; auto; left.
```

```
    rewrite H9, H10; unfold Less; auto.
  - assert (Ordinal b /\ Ordinal d); auto; apply Theorem110 in H12.
    destruct H12 as [H12 | [H12 | H12]].
    { left; unfold Rrelation, LessLess; apply AxiomII.
      split; try (apply Theorem49; split; apply Theorem49; auto).
      exists a, b, c, d; repeat split; auto; right; right.
      rewrite H9, H10; unfold Less; split; auto. }
    { right; left; unfold Rrelation, LessLess; apply AxiomII.
      split; try (apply Theorem49; split; apply Theorem49; auto).
      exists c, d, a, b; repeat split; auto; right; right.
      rewrite H9, H10; unfold Less; split; auto. }
    { right; right; rewrite H11, H12; auto. }
Qed.

Lemma Lemma177_v : ∀ x y u v,
  x ∈ y -> y ⊂ (R × R) -> Max u v = v -> Rrelation x ≪ ([u,v]) ->
  (∃ a b, [a, b] ∈ y /\ Ensemble a /\ Ordinal a /\ Ensemble b /\
  Ordinal b /\ ((Max a b) ∈ v \/ Max a b=v /\ a ∈ u \/ Max a b = v /\
  a = u /\ b ∈ v)).
Proof.
  intros.
  double H; apply H0 in H3; unfold Cartesian in H3; clear H0.
  PP H3 a b; apply AxiomII_P in H0; clear H3; exists a, b; split; auto.
  destruct H0, H3; apply AxiomII in H3; apply AxiomII in H4; clear H0.
  destruct H3, H4; unfold Rrelation, LessLess in H2;
  apply AxiomII in H2.
  destruct H2, H6, H6, H6, H6, H6, H7, H8; clear H6 H7;
  apply Theorem49 in H2.
  destruct H2; clear H2; apply Theorem49 in H6; destruct H6;
  unfold Less in H9.
  apply Theorem55 in H8; try (split; Ens; apply Theorem49; Ens).
  destruct H8; apply Theorem55 in H7; apply Theorem55 in H8; auto.
  destruct H7, H8; rewrite <- H7, <- H8, <- H10, <- H11, H1 in H9; Ens.
Qed.

Lemma Lemma177_u : ∀ x y u v,
  x ∈ y -> y ⊂ (R × R) -> Max u v = u -> Rrelation x ≪ ([u,v]) ->
  (∃ a b, [a, b] ∈ y /\ Ensemble a /\ Ordinal a /\ Ensemble b /\
  Ordinal b /\ ((Max a b) ∈ u \/ Max a b=u /\ a ∈ u \/ Max a b = u /\
  a = u /\ b ∈ v)).
Proof.
  intros.
  double H; apply H0 in H3; unfold Cartesian in H3; clear H0.
  PP H3 a b; apply AxiomII_P in H0; clear H3; exists a, b; split; auto.
  destruct H0, H3; apply AxiomII in H3; apply AxiomII in H4; clear H0.
```

```
  destruct H3, H4; unfold Rrelation, LessLess in H2;
  apply AxiomII in H2.
  destruct H2, H6, H6, H6, H6, H6, H7, H8; clear H6 H7;
  apply Theorem49 in H2.
  destruct H2; clear H2; apply Theorem49 in H6;
  destruct H6; unfold Less in H9.
  apply Theorem55 in H8; try (split; Ens; apply Theorem49; Ens).
  destruct H8; apply Theorem55 in H7; apply Theorem55 in H8; auto.
  destruct H7,H8;rewrite <- H7, <- H8, <- H10, <- H11,H1 in H9; Ens.
Qed.

Theorem Theorem177 : WellOrdered ≪ (R × R).
Proof.
  unfold WellOrdered; split; intros.
  - unfold Connect;intros; destruct H; double H;double H0; PP H1 a b.
    PP H2 c d;clear H1 H2;apply AxiomII_P in H3;apply AxiomII_P in H4.
    destruct H3, H4; clear H1 H3; destruct H2, H4;
    unfold R in H1, H2, H3, H4.
    apply AxiomII in H1; apply AxiomII in H2; apply AxiomII in H3.
    apply AxiomII in H4; destruct H1, H2, H3, H4.
    assert (Ordinal a /\ Ordinal b);
    assert (Ordinal c /\ Ordinal d); auto.
    apply Theorem110 in H9; apply Theorem110 in H10.
    destruct H9 as [H9 | [H9 | H9]], H10 as [H10 | [H10 | H10]];
    apply Property_Max in H9; apply Property_Max in H10; auto.
    + apply Lemma177_bd; auto.
    + rewrite Equal_Max in H10; apply Lemma177_bc; auto.
    + apply Lemma177_bd; auto.
    + rewrite Equal_Max in H9;apply (Lemma177_bc c d a b) in H10;auto.
      destruct H10 as [H10 | [H10| H10]]; try rewrite H10; auto.
    + rewrite Equal_Max in H9, H10; apply Lemma177_ac; auto.
    + rewrite Equal_Max in H9;apply (Lemma177_bc c d a b) in H10;auto.
      destruct H10 as [H10 | [H10| H10]]; try rewrite H10; auto.
    + apply Lemma177_bd; auto.
    + rewrite Equal_Max in H10; apply Lemma177_bc; auto.
    + apply Lemma177_bd; auto.
  - destruct H.
    assert ((En_y y) ⊂ R /\ (En_y y) ≠ ∅).
    { split.
      - unfold Included; intros; apply AxiomII in H1;
        destruct H1, H2, H2, H2.
        apply H in H2;apply AxiomII_P in H2;destruct H2,H4;clear H2.
        apply AxiomII in H4; apply AxiomII in H5; destruct H4, H5.
        assert (Ordinal x /\ Ordinal x0); auto; apply Theorem110 in H7.
        rewrite H3; destruct H7 as [H7|[H7|H7]];
```

```
    apply Property_Max in H7; auto;
    try (rewrite H7; unfold R; apply AxiomII; Ens).
    rewrite Equal_Max in H7; rewrite H7; apply AxiomII; Ens.
  - apply Property_NotEmpty in H0; destruct H0; double H0;
    apply H in H1; PP H1 a b.
    apply AxiomII_P in H2; clear H1; destruct H2, H2; clear H1.
    apply AxiomII in H2; apply AxiomII in H3; destruct H2, H3;
    apply Property_NotEmpty.
    assert (Ordinal a /\ Ordinal b); auto; apply Theorem110 in H5.
    destruct H5 as [H5|[H5|H5]]; apply Property_Max in H5; auto.
    + exists b; apply AxiomII; split; auto; exists a, b; auto.
    + exists a; apply AxiomII; split; auto; exists a, b;split;auto.
      rewrite Equal_Max; symmetry; auto.
    + exists b; apply AxiomII; split; auto; exists a, b; auto. }
clear H0; apply Lemma121 in H1;
unfold FirstMember in H1; destruct H1.
apply AxiomII in H0; destruct H0, H2 as [u [v H2]], H2.
generalize (classic ((En_v (⋂ En_y y) y) = ∅));intros; destruct H4.
+ generalize (classic ((En_u (⋂ En_y y) y) = ∅)); intros;
  destruct H5.
  * double H2; apply H in H6; unfold Cartesian in H6;
    apply AxiomII_P in H6.
    destruct H6, H7; apply AxiomII in H7;
    apply AxiomII in H8; clear H6.
    destruct H7, H8; assert (Ordinal u /\ Ordinal v); auto.
    apply Theorem110 in H10; destruct H10 as [H10 | [H10 | H10]].
    { double H10; apply Property_Max in H11; auto.
      rewrite H11 in H3; rewrite H3 in *; clear H3 H11.
      assert (u ∈ (En_v v y)); try (apply AxiomII; Ens);
      rewrite H4 in H3.
      generalize (Theorem16 u); intros; contradiction. }
    { double H10; apply Property_Max in H11; auto;
      rewrite Equal_Max in H11.
      rewrite H11 in H3; rewrite H3 in *; clear H3 H11.
      assert (v ∈ (En_u u y)); try (apply AxiomII; Ens);
      rewrite H5 in H3.
      generalize (Theorem16 v); intros; contradiction. }
    { double H10; apply Property_Max in H11; rewrite H11 in H3.
      rewrite H3, H10 in *; clear H3 H6 H8 H9 H10; exists [v,v].
      unfold FirstMember; split; auto; intros; intro.
      apply Lemma177_v with (y:= y) in H6; auto; clear H3 H11.
      destruct H6 as [a [b H6]], H6, H6, H8, H9, H10.
      assert(Ordinal a /\ Ordinal b);auto;apply Theorem110 in H12.
      destruct H12 as [H12 | [H12 | H12]].
      - apply Property_Max in H12; auto; rewrite H12 in H11.
```

```
      destruct H11 as [H11 | [H11 | H11]].
      + assert (b ∈ (En_y y)); try apply AxiomII; Ens.
        apply H1 in H13; elim H13; clear H12 H13;
        unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      + destruct H11; rewrite H11 in *; clear H9 H10 H11.
        assert (a ∈ (En_v v y)); try apply AxiomII; Ens.
        rewrite H4 in H9; generalize (Theorem16 a); contradiction.
      + destruct H11, H13; rewrite H11 in H14.
        generalize (Theorem101 v); intros; contradiction.
    - apply Property_Max in H12; auto; rewrite Equal_Max in H12.
      rewrite H12 in H11; destruct H11 as [H11 | [H11 | H11]].
      + assert (a ∈ (En_y y)); try apply AxiomII; Ens.
        apply H1 in H13; elim H13; clear H12 H13;
        unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      + destruct H11; rewrite H11 in H13.
        generalize (Theorem101 v); intros; contradiction.
      + destruct H11; clear H11; destruct H13; rewrite H11 in *.
        clear H11 H12; assert (b ∈ (En_u v y));
        try apply AxiomII; Ens.
        rewrite H5 in H11;generalize(Theorem16 b);contradiction.
    - double H12; apply Property_Max in H13; auto;
      rewrite H13 in H11.
      rewrite H12 in *; clear H9 H10 H12;
      destruct H11 as [H9|[H9| H9]].
      + assert (b ∈ (En_y y)); try apply AxiomII; Ens.
        apply H1 in H10; elim H10; clear H13 H10;
        unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      + destruct H9; rewrite H9 in H10.
        generalize (Theorem101 v); intros; contradiction.
      + destruct H9, H10; rewrite H9 in H11.
        generalize (Theorem101 v); intros; contradiction. }
* assert ((En_u (⋂ En_y y) y) ⊂ R /\ (En_u (⋂ En_y y) y) ≠ ∅).
  { split; auto; unfold Included; intros; apply AxiomII in H6.
    destruct H6, H7; apply H in H7; apply AxiomII_P in H7;
    apply H7. }
  apply Lemma121 in H6;clear H5;destruct H6;apply AxiomII in H5.
  destruct H5, H7; exists [⋂ (En_y y), ⋂ (En_u (⋂(En_y y)) y)].
  clear H5; double H7; apply H in H7; apply AxiomII_P in H7.
  destruct H7; clear H7; destruct H9; apply AxiomII in H7.
  apply AxiomII in H9; clear H0; destruct H7, H9.
  double H8; apply Property_Max in H11; auto.
  unfold FirstMember; split; auto; intros; intro.
```

```
    apply Lemma177_u with (y:=y) in H13;try rewrite Equal_Max;auto.
    clear H12; destruct H13 as [a [b H13]], H13,H13,H14,H15,H16.
    assert (Ordinal a /\ Ordinal b); auto; apply Theorem110 in H18.
    destruct H18 as [H18 | [H18 | H18]].
    { double H18;apply Property_Max in H19;auto;rewrite H19 in H17.
      destruct H17 as [H17 | [H17 | H17]].
      - assert (b ∈ (En_y y)); try apply AxiomII; Ens.
        apply H1 in H20;elim H20;clear H19 H20;unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      - destruct H17; rewrite H17 in *; clear H15 H16 H17.
        assert (a ∈ (En_v (⋂ En_y y) y)); try apply AxiomII; Ens.
        rewrite H4 in H15; generalize (Theorem16 a); contradiction.
      - destruct H17, H20; rewrite<-H17 in H20; rewrite H20 in H18.
        generalize (Theorem101 b); intros; contradiction. }
    { double H18; apply Property_Max in H19; auto;
      rewrite Equal_Max in H19.
      rewrite H19 in H17; destruct H17 as [H17 | [H17 | H17]].
      - assert (a ∈ (En_y y)); try apply AxiomII; Ens.
        apply H1 in H20; elim H20; clear H19 H20;
        unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      - destruct H17; rewrite <- H17 in H20.
        generalize (Theorem101 a); intros; contradiction.
      - destruct H17; clear H17; destruct H20; rewrite H17 in *.
        assert (b∈(En_u (⋂ En_y y) y)); try apply AxiomII; Ens.
        apply H6 in H21. elim H21; unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto. }
    { double H18; apply Property_Max in H19; rewrite H19 in H17.
      rewrite H18 in *; clear H15 H16 H18;
      destruct H17 as [H15|[H15| H15]].
      - assert (b ∈ (En_y y)); try apply AxiomII; Ens.
        apply H1 in H16;elim H16;clear H19 H16;unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      - destruct H15; rewrite <- H15 in H16.
        generalize (Theorem101 b); intros; contradiction.
      - destruct H15; clear H15; destruct H16; rewrite H15 in H16.
        generalize (Theorem102 (⋂ En_y y) (⋂ En_u (⋂ En_y y) y));
        intros.
        destruct H17; split; auto. }
+ assert ((En_v (⋂ En_y y) y) ⊂ R /\ (En_v (⋂ En_y y) y) ≠ ∅).
  { split; auto; unfold Included; intros; apply AxiomII in H5.
    destruct H5, H6; apply H in H6;
    apply AxiomII_P in H6; apply H6.}
  apply Lemma121 in H5; clear H4; destruct H5; apply AxiomII in H4.
  destruct H4, H6; exists [⋂ (En_v (⋂ En_y y) y), ⋂ (En_y y)].
```

```
    clear H4; double H6; apply H in H6; apply AxiomII_P in H6;
    destruct H6.
    clear H6; destruct H8; apply AxiomII in H6; apply AxiomII in H8.
    clear H0; destruct H6, H8; double H7;
    apply Property_Max in H10; auto.
    unfold FirstMember; split; auto; intros; intro.
    apply Lemma177_v with (y:= y) in H12; auto; clear H11.
    destruct H12 as [a [b H12]], H12, H12, H13, H14, H15.
    assert (Ordinal a /\ Ordinal b); auto; apply Theorem110 in H17.
    destruct H17 as [H17 | [H17 | H17]].
    { double H17; apply Property_Max in H18; auto; rewrite H18 in H16.
      destruct H16 as [H16 | [H16 | H16]].
      - assert (b ∈ (En_y y)); try apply AxiomII; Ens.
        apply H1 in H19; elim H19; clear H18 H19; unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      - destruct H16; rewrite H16 in *; clear H14 H15 H16.
        assert (a ∈ (En_v (⋂ En_y y) y)); try apply AxiomII; Ens.
        apply H5 in H14; destruct H14; unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      - destruct H16, H19; rewrite <- H16 in H20.
        generalize (Theorem101 b); intros; contradiction. }
    { double H17; apply Property_Max in H18; auto;
      rewrite Equal_Max in H18.
      rewrite H18 in H16; destruct H16 as [H16 | [H16 | H16]].
      - assert (a ∈ (En_y y)); try apply AxiomII; Ens.
        apply H1 in H19; destruct H19; unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      - destruct H16; rewrite H16 in H19.
        generalize (Theorem102 (⋂ En_y y) (⋂ En_v (⋂ En_y y) y));
        intros.
        destruct H20; split; auto.
      - destruct H16, H19; rewrite <- H19, <- H16 in H7.
        generalize (Theorem101 a); intros; contradiction. }
    { double H17; apply Property_Max in H18; rewrite H18 in H16.
      rewrite H17 in *; clear H14 H15 H17;
      destruct H16 as [H14|[H14| H14]].
      - assert (b ∈ (En_y y)); try apply AxiomII; Ens.
        apply H1 in H15; destruct H15; unfold Rrelation, E.
        apply AxiomII_P; split; try apply Theorem49; auto.
      - destruct H14; rewrite <- H14 in *.
        generalize (Theorem102 b (⋂ En_v b y)); intros;
        destruct H16; auto.
      - destruct H14, H15; rewrite <- H15, <- H14 in H7.
        generalize (Theorem101 b); intros; contradiction. }
Qed.
```

```
Hint Resolve Theorem177 : set.
```

定理 178 $(u,v) \ll (x,y) \Longrightarrow (u,v) \in (\max[x,y]+1) \times (\max[x,y]+1)$.

```
Theorem Theorem178 : ∀ u v x y,
  Rrelation ([u,v]) ≪ ([x,y]) ->
  [u,v] ∈ ((PlusOne (Max x y)) × (PlusOne (Max x y))).
Proof.
  intros.
  unfold Rrelation, LessLess in H; apply AxiomII in H.
  destruct H, H0, H0, H0, H0, H0, H1, H2; apply Theorem49 in H;
  destruct H.
  apply Theorem55 in H2; auto; destruct H2; apply Theorem49 in H.
  apply Theorem49 in H4; destruct H, H4; apply Theorem55 in H2; auto.
  apply Theorem55 in H5; auto; destruct H2, H5;
  rewrite <- H2, <- H5 in *.
  rewrite <- H8, <- H9 in *; clear H H2 H4 H5 H6 H7 H8 H9 x0 x1 x2 x3.
  assert ((Max u v) ≪ (Max x y)).
  { unfold LessEqual; destruct H3 as [H3|[H3|H3]]; try tauto. }
  clear H3; unfold Cartesian in H0, H1; apply AxiomII_P in H0.
  apply AxiomII_P in H1; destruct H0, H1, H2, H3;
  unfold R in H2, H3, H4, H5.
  clear H0 H1; apply AxiomII in H2; apply AxiomII in H3;
  apply AxiomII in H4.
  apply AxiomII in H5; destruct H2, H3, H4, H5, H.
  - assert ((Max x y) ∈ R).
    { assert (Ordinal x /\ Ordinal y); auto; apply Theorem110 in H8.
      destruct H8 as[H8|[H8|H8]]; apply Property_Max in H8; auto;
      try rewrite H8.
      - unfold R; apply AxiomII; split; auto.
      - rewrite Equal_Max in H8; rewrite H8; unfold R;
        apply AxiomII; auto.
      - unfold R; apply AxiomII; split; auto. }
    double H8; apply Lemma123 in H9; unfold R in H8, H9;
    apply AxiomII in H8.
    apply AxiomII in H9; destruct H8, H9; unfold LessEqual in H.
    assert ((Max x y) ∈ (PlusOne (Max x y))).
    { unfold PlusOne; apply Theorem4; right;
      apply AxiomII; split; auto. }
    unfold Ordinal, full in H11; destruct H11 as [_ H11];
    apply H11 in H12.
    apply H12 in H; clear H8 H9 H10 H12;
    assert (Ordinal u /\ Ordinal v); auto.
    apply Theorem110 in H8; destruct H8 as [H8 | [H8 | H8]].
```

```
    + double H8; apply Property_Max in H9; auto;
      rewrite H9 in H; clear H9.
      double H; apply H11 in H9; apply H9 in H8; clear H9 H11;
      unfold Cartesian.
      apply AxiomII_P; repeat split; try apply Theorem49; auto.
    + double H8; apply Property_Max in H9; auto;
      rewrite Equal_Max in H9.
      rewrite H9 in H; clear H9; double H; apply H11 in H9;
      apply H9 in H8.
      clear H9 H11; unfold Cartesian; apply AxiomII_P.
      repeat split; try apply Theorem49; auto.
    + double H8; apply Property_Max in H9; auto; rewrite H9 in H;
      rewrite H8.
      clear H8 H9 H11; apply AxiomII_P; repeat split;
      try apply Theorem49; auto.
  - rewrite <- H in *; clear H; assert (Ordinal u /\ Ordinal v); auto.
    apply Theorem110 in H; destruct H as [H | [H | H]].
    + double H; apply Property_Max in H8; auto; rewrite H8; clear H8.
      unfold Cartesian; apply AxiomII_P; repeat split;
      try apply Theorem49; auto.
      * apply Theorem4; tauto.
      * apply Theorem4; right; apply AxiomII; auto.
    + double H; apply Property_Max in H8; auto; rewrite Equal_Max in H8.
      rewrite H8; clear H8; unfold Cartesian; apply AxiomII_P.
      repeat split; try apply Theorem49; auto;
      try (apply Theorem4; tauto).
      apply Theorem4; right; apply AxiomII; auto.
    + double H; apply Property_Max in H8; rewrite H8; clear H8;
      rewrite H at 1.
      unfold Cartesian; apply AxiomII_P; split;
      try apply Theorem49; auto.
      split; apply Theorem4; right; apply AxiomII; auto.
Qed.

Hint Resolve Theorem178 : set.
```

定理 179　$x \in C \sim \omega \Longrightarrow P(x \times x) = P(x)$.

```
Definition En_Q u v x0 : Class :=
  \{\ λ a b, Rrelation ([a,b]) ≪ ([u,v]) /\ [a,b] ∈ x0 × x0 \}\.

Lemma Lemma179 : ∀ x, Ensemble x -> P[x × x] =
  P[(P[x]) × (P[x])].
Proof.
  intros.
```

```
double H; double H; apply Property_PClass in H0.
apply Theorem153 in H1; apply Theorem146 in H1.
unfold Equivalent in H1; destruct H1 as [f H1], H1, H2.
assert (Ensemble (x × x) /\ Ensemble ((P[x]) × (P[x]))).
{ split; apply Theorem74; Ens. }
apply Theorem154 in H4; apply H4; clear H4.
unfold Equivalent.
exists \{\ λ a b, a ∈ (x × x) /\ b = [f[First a], f[Second a]] \}\.
repeat split; unfold Relation; intros; try PP H4 c d; Ens.
- destruct H4; apply AxiomII_P in H4; apply AxiomII_P in H5.
  destruct H4, H5, H6, H7; rewrite H8, H9; auto.
- destruct H4; apply AxiomII_P in H4; apply AxiomII_P in H5.
  destruct H4, H5; apply AxiomII_P in H6; apply AxiomII_P in H7.
  destruct H6, H7, H8, H9; rewrite H11 in H10; clear H4 H5 H6 H7 H11.
  PP H8 a b;PP H9 c d;clear H8 H9;apply AxiomII_P in H4;clear y z.
  apply AxiomII_P in H5; destruct H4, H5, H6, H7;
  apply Theorem49 in H4.
  apply Theorem49 in H5;apply Theorem54 in H4;apply Theorem54 in H5.
  destruct H4, H5; rewrite H4, H5, H11, H12 in H10;
  clear H4 H5 H11 H12.
  rewrite <- H2 in H6, H7, H8, H9; destruct H1.
  apply Property_Value in H6; apply Property_Value in H7; auto.
  apply Property_Value in H8; apply Property_Value in H9; auto.
  double H7; double H9; apply Property_ran in H7;
  apply Property_ran in H11.
  AssE f[c]; AssE f[d]; apply Theorem55 in H10; auto;
  clear H7 H11 H12 H13.
  destruct H10; rewrite H7 in H5; rewrite H10 in H9; clear H7 H10.
  assert ([f[a],a] ∈ f⁻¹ /\ [f[a],c] ∈ f⁻¹).
  { AssE [a,f[a]]; AssE [c,f[a]]; apply Theorem49 in H7; destruct H7.
    apply Theorem49 in H10; destruct H10 as [H10 _]; unfold Inverse.
    split; apply AxiomII_P; split; try apply Theorem49; auto. }
  assert ([f[b],b] ∈ f⁻¹ /\ [f[b],d] ∈ f⁻¹).
  { AssE [b,f[b]]; AssE [d,f[b]];apply Theorem49 in H10;destruct H10.
    apply Theorem49 in H11; destruct H11 as [H11 _]; unfold Inverse.
    split; apply AxiomII_P; split; try apply Theorem49; auto. }
  apply H4 in H7; apply H4 in H10; rewrite H7, H10; auto.
- apply AxiomI; split; intros.
  + apply AxiomII in H4; destruct H4, H5;
    apply AxiomII_P in H5; apply H5.
  + apply AxiomII; split; Ens; PP H4 a b; exists [f[a],f[b]].
    double H5; apply AxiomII_P in H6; rewrite <- H2 in H6;
    destruct H6, H7.
    destruct H1; apply Property_Value in H7;
    apply Property_Value in H8; Ens.
```

```
      apply Property_ran in H7; apply Property_ran in H8.
      apply AxiomII_P; repeat split; try apply Theorem49;
      try split; auto.
      * apply Theorem49; Ens.
      * apply Theorem49 in H6; apply Theorem54 in H6; destruct H6.
        rewrite H6, H10; auto.
  - apply AxiomI; split; intros.
    + apply AxiomII in H4; destruct H4, H5; apply AxiomII_P in H5.
      destruct H5, H6; PP H6 a b; AssE [a,b]; apply Theorem49 in H9.
      apply Theorem54 in H9; destruct H9; rewrite H9, H10 in H7;
      clear H9 H10.
      rewrite H7 in *; clear H5 H6 H7; apply AxiomII_P in H8;
      destruct H8,H6.
      destruct H1; rewrite <- H2 in H6, H7;
      apply Property_Value in H6; auto.
      apply Property_Value in H7; auto; apply Property_ran in H6.
      apply Property_ran in H7; rewrite H3 in H6, H7; unfold Cartesian.
      apply AxiomII_P; repeat split; auto.
    + apply AxiomII; split; Ens; PP H4 a b; clear H4;
      apply AxiomII_P in H5.
      destruct H5, H5; rewrite <- H3 in H5, H6; unfold Range in H5, H6.
      apply AxiomII in H5;apply AxiomII in H6;destruct H5, H6, H7, H8.
      double H7; double H8; apply Property_dom in H9;
      apply Property_dom in H10.
      exists [x0,x1]; assert (Ensemble ([x0,x1]));
      try apply Theorem49; Ens.
      apply AxiomII_P; split; try apply Theorem49; split; auto.
      * rewrite H2 in H9, H10; unfold Cartesian; apply AxiomII_P.
        repeat split; try apply Theorem49; Ens.
      * apply Theorem49 in H11; apply Theorem54 in H11;
        destruct H1, H11.
        rewrite H11, H13; clear H11 H13;
        apply Property_Value in H9; auto.
        apply Property_Value in H10; auto; add ([x0, a] ∈ f) H9.
        add ([x1, b] ∈ f) H10; apply H1 in H9; apply H1 in H10.
        rewrite H9, H10; auto.
Qed.

Lemma Lemma179' : ∀ x x0, Ensemble x0 -> x ≈ x × [x0].
Proof.
  intros.
  unfold Equivalent; exists (\{\ λ a b, a ∈ x /\ b = [a,x0] \}\).
  repeat split; intros; try (unfold Relation; intros; PP H0 a b; Ens).
  - destruct H0; apply AxiomII_P in H0; apply AxiomII_P in H1.
    destruct H0, H1, H2, H3; rewrite H4, H5; auto.
```

```
  - destruct H0; apply AxiomII_P in H0; apply AxiomII_P in H1.
    destruct H0, H1; apply AxiomII_P in H2; apply AxiomII_P in H3.
    destruct H2, H3, H4, H5; apply Theorem49 in H0; destruct H0.
    rewrite H6 in H7; apply Theorem55 in H7; try apply H7; auto.
  - apply AxiomI; split; intros.
    + unfold Domain in H0; apply AxiomII in H0; destruct H0, H1.
      apply AxiomII_P in H1; apply H1.
    + unfold Domain; apply AxiomII; split; Ens; exists [z,x0].
      apply AxiomII_P; repeat split; auto; apply Theorem49; split; Ens.
      apply Theorem49; split; Ens.
  - apply AxiomI; split; intros.
    + unfold Range in H0; apply AxiomII in H0; destruct H0, H1.
      apply AxiomII_P in H1; destruct H1, H2; rewrite H3 in *.
      unfold Cartesian; apply AxiomII_P; repeat split; auto.
      unfold Singleton; apply AxiomII; split; Ens.
    + unfold Range; apply AxiomII; split; Ens; PP H0 a b.
      apply AxiomII_P in H1; destruct H1, H2; exists a.
      apply AxiomII_P; repeat split; auto; try apply Theorem49; Ens.
      apply Theorem49 in H1; apply Theorem55; auto; split; auto.
      unfold Singleton in H3; apply AxiomII in H3; destruct H3.
      apply H4; apply Theorem19; Ens.
Qed.

Lemma Lemma179'' : ∀ f u v x0 ,
  Ensemble f[[u,v]] -> Ordinal f[[u, v]] -> x0 ∈ C ->
  P[f[[u, v]]] ∈ x0 -> f[[u, v]] ∈ x0.
Proof.
  intros; double H1.
  apply Theorem156 in H1; destruct H1; unfold C in H3.
  apply AxiomII in H3; destruct H3 as [_ H3];
  unfold Cardinal_Number in H3.
  destruct H3 as [H3 _]; apply AxiomII in H3; destruct H3 as [_ H3].
  assert (Ordinal f[[u, v]] /\ Ordinal x0); auto.
  apply Theorem110 in H5; destruct H5 as [H5 | [H5 | H5]]; auto.
  - unfold Ordinal in H0; destruct H0 as [_ H0]; apply H0 in H5.
    add (x0 ⊂ f [[u, v]]) H; apply Theorem158 in H; clear H5.
    rewrite H4 in H; unfold LessEqual in H; destruct H.
    + generalize (Theorem102 x0 P[f[[u,v]]]); intros; destruct H5; auto.
    + rewrite <- H in H2; generalize (Theorem101 x0); contradiction.
  - rewrite H5, H4 in H2; generalize (Theorem101 x0); contradiction.
Qed.

Theorem Theorem179 : ∀ x, x ∈ (C ∼ ω) -> P[x × x] = x.
Proof.
  intros.
```

```
generalize Theorem150; intros.
unfold WellOrdered in H0; destruct H0; clear H0.
generalize (classic (\{ λ z, z ∈ (C ~ ω) /\ P[z × z] <> z \} = ∅));
intros.
destruct H0.
- generalize (classic (P[x × x] = x)); intros; destruct H2; auto.
  assert (x ∈ ∅). { rewrite <- H0; apply AxiomII; Ens. }
  generalize (Theorem16 x); intros; contradiction.
- assert (\{ λ z, z ∈ (C ~ ω) /\ P[z × z] <> z \} ⊂ C /\
          \{ λ z, z ∈ (C ~ ω) /\ P[z × z] <> z \} ≠ ∅).
  { split; auto; unfold Included; intros; apply AxiomII in H2.
    destruct H2, H3; apply Theorem4' in H3; apply H3. }
  apply H1 in H2; clear H0 H1; destruct H2; unfold FirstMember in H0.
  destruct H0; apply AxiomII in H0; destruct H0, H2.
  generalize Theorem177, Theorem113; intros; destruct H5.
  assert (Ensemble x0 /\ Ensemble x0); Ens; apply Theorem74 in H7.
  assert (x0 × x0 ⊂ R × R).
  { unfold Setminus in H2; apply Theorem4' in H2;
    destruct H2 as [H2 _].
    unfold C in H2; apply AxiomII in H2; destruct H2 as [_ H2].
    destruct H2 as [H2 _]; unfold Ordinal_Number in H2;
    unfold Ordinal in H5.
    destruct H5 as [_ H5]; apply H5 in H2; clear H5.
    unfold Cartesian, Included; intros; PP H5 a b;
    apply AxiomII_P in H8.
    destruct H8, H9; apply H2 in H9; apply H2 in H10;
    apply AxiomII_P; Ens. }
  apply Lemma97 with (y:= x0 × x0) in H4; auto; clear H8.
  apply Theorem107 in H5; add (WellOrdered E R) H4; auto; clear H5.
  apply Theorem100 in H4; auto; clear H6 H7;
  destruct H4 as [f H4], H4, H5.
  unfold Order_PXY in H5; destruct H5, H7, H8;
  clear H4 H5 H7; double H8.
  apply Theorem96 in H4; destruct H4 as [H4 _], H9.
  assert (forall u v, [u,v] ∈ (x0 × x0) -> f[[u,v]] ∈ x0).
  { intros.
    assert ((En_Q u v x0) ⊂
    ((PlusOne (Max u v)) × (PlusOne (Max u v)))).
    { unfold Included; intros; PP H10 a b; apply AxiomII_P in H11.
      destruct H11, H12; apply Theorem178; auto. }
    assert (En_Q u v x0 ≈ f[[u, v]]).
    { rewrite <- H6 in H9; apply Property_Value in H9; try apply H4.
      apply Property_ran in H9; clear H10; unfold Equivalent.
      exists (f|(En_Q u v x0)); destruct H4; double H4; double H10.
      apply (Theorem126 f (En_Q u v x0)) in H11; destruct H11, H13.
```

```
apply (Theorem126 f⁻¹ f[[u, v]]) in H12; destruct H12, H15.
split; try split; auto; unfold Function, Relation.
- split; intros; try PP H17 a b; Ens; unfold Inverse in H17.
  destruct H17; apply AxiomII_P in H17; apply AxiomII_P in H18.
  destruct H17, H18; unfold Restriction in H19, H20.
  apply Theorem4' in H19; apply Theorem4' in H20.
  destruct H19 as [H19 _], H20 as [H20 _].
  assert ([x1,z] ∈ f⁻¹ /\ [x1,y] ∈ f⁻¹).
  { unfold Inverse; split; apply AxiomII_P; auto. }
  apply H10 in H21; symmetry; auto.
- assert (En_Q u v x0 ⊂ dom(f)).
  { unfold Included, En_Q; intros; PP H17 a b; rewrite H6.
    apply AxiomII_P in H18; apply H18. }
  apply Theorem30 in H17; rewrite H17 in H13; auto.
- apply AxiomI; split; intros.
  + unfold Range in H17; apply AxiomII in H17; destruct H17, H18.
    unfold Restriction in H18; apply Theorem4' in H18;
    destruct H18.
    unfold Cartesian in H19; apply AxiomII_P in H19.
    destruct H19 as [_ H19], H19 as [H19 _];
    PP H19 a b; clear H19.
    apply AxiomII_P in H20; destruct H20, H20;
    rewrite <- H6 in H21.
    double H21; apply Property_Value in H22; auto.
    add ([[a, b], z] ∈ f) H22; clear H18 H19; apply H4 in H22.
    rewrite <- H22; clear H22;
    apply Property_Value' in H9; auto.
    apply Property_dom in H9; unfold Order_Pr in H5.
    assert ([a,b] ∈ dom( f) /\ [u,v] ∈ dom( f) /\
    Rrelation ([a,b]) ≪ ([u,v])); auto.
    apply H5 in H18; unfold Rrelation, E in H18.
    apply AxiomII_P in H18; apply H18.
  + unfold Range; apply AxiomII; split; Ens;
    double H8; double H9.
    apply Theorem114 in H18; destruct H18 as [_ H18].
    apply H18 in H19; double H17;
    apply H19 in H20; clear H19 H18.
    double H20; apply AxiomII in H19; destruct H19, H20;
    exists x1.
    unfold Restriction; apply Theorem4'; split; auto.
    unfold Cartesian; apply AxiomII_P; split; Ens.
    split; try apply Theorem19; Ens;
    clear H H1 H2 H3 H13 H14 H15 H16.
    apply Theorem96 in H5; destruct H5 as [_ H5].
    unfold Order_Pr in H5; rewrite <- Lemma96' in H5.
```

```
      assert (z∈ran(f) /\ f[[u,v]]∈ran(f) /\
      Rrelation z E f[[u, v]]).
      { repeat split; auto; unfold Rrelation, E; apply AxiomII_P.
        split; auto; apply Theorem49; Ens. }
      apply H5 in H; pattern f at 3 in H;
      rewrite <- Theorem61 in H; try apply H4.
      apply Property_Value' in H9; auto; apply Property_dom in H9.
      rewrite Lemma96 in H9; double H20; apply Property_ran in H2.
      rewrite <- Lemma96''' in H; try rewrite Theorem61;
      try apply H4; auto.
      rewrite Lemma96' in H2; apply Property_Value in H2; auto.
      assert ([z,x1] ∈ f ⁻¹).
      { apply AxiomII_P; split; auto; apply Theorem49.
        AssE [x1,z]; apply Theorem49 in H3; destruct H3; auto. }
      add ([z,x1] ∈ f ⁻¹) H2; apply H10 in H2; rewrite H2 in H.
      apply Property_dom in H1; rewrite H6 in H1; clear H2 H3.
      PP H1 a b; unfold En_Q;apply AxiomII_P;repeat split;Ens.}
assert ([u,v] ∈ (ω × ω) -> f[[u,v]] ∈ x0).
{ clear H9; intros; clear H x.
  assert (ω × ω ⊂ x0 × x0).
  { unfold Included; intros; PP H a b; apply AxiomII_P in H12.
    destruct H12, H13; double H13; double H14;
    unfold ω in H15, H16.
    apply AxiomII in H15; apply AxiomII in H16;
    destruct H15 as [_ H15].
    destruct H16 as [_ H16], H15 as [H15 _], H16 as [H16 _].
    apply AxiomII_P; split; auto; apply Theorem4' in H2;
    destruct H2.
    unfold C in H2; apply AxiomII in H2; destruct H2 as [_ H2].
    unfold Cardinal_Number, Ordinal_Number in H2;
    destruct H2 as[H2 _].
    apply AxiomII in H2; destruct H2 as [_ H2];
    apply AxiomII in H17.
    destruct H17 as [_ H17]; add (Ordinal x0) H15;
    add (Ordinal x0) H16.
    apply Theorem110 in H15; apply Theorem110 in H16.
    destruct H15 as [H15|[H15|H15]],H16 as [H16|[H16|H16]];auto.
    - destruct H17; apply AxiomII; split; Ens.
      apply (Theorem132 b _); auto; apply AxiomII in H14;
      apply H14.
    - rewrite H16 in H14; contradiction.
    - destruct H17; apply AxiomII; split; Ens.
      apply (Theorem132 a _); auto; apply AxiomII in H13;
      apply H13.
    - destruct H17; apply AxiomII; split; Ens.
```

```
      apply (Theorem132 b _); auto; apply AxiomII in H14;
      apply H14.
    - rewrite H16 in H14; contradiction.
    - rewrite H15 in H13; contradiction.
    - rewrite H15 in H13; contradiction.
    - rewrite H15 in H13; contradiction. }
double H9; apply H in H9; rewrite <- H6 in H9.
apply AxiomII_P in H12; destruct H12, H13.
assert (PlusOne (Max u v) ∈ ω).
{ apply Theorem134; double H13; double H14; unfold ω in H15, H16.
  apply AxiomII in H15; apply AxiomII in H16;
  destruct H15 as [_ H15].
  destruct H16 as [_ H16], H15 as [H15 _], H16 as [H16 _].
  assert (Ordinal u /\ Ordinal v); auto; apply Theorem110 in H17.
  destruct H17 as [H17|[H17|H17]];
  try apply Property_Max in H17; auto.
  - rewrite H17; auto.
  - rewrite Equal_Max in H17; rewrite H17; auto.
  - rewrite H17; auto. }
assert(Finite(PlusOne(Max u v)) /\ Finite (PlusOne (Max u v))).
{ double H15; generalize Theorem164; intros; apply H17 in H16.
  clear H17; apply Theorem156 in H16; destruct H16 as [_ H16].
  unfold Finite; rewrite H16; auto. }
apply Theorem170 in H16; unfold Finite in H16.
assert (Ensemble ((PlusOne (Max u v)) × (PlusOne (Max u v))) /\
((En_Q u v x0) ⊂ (PlusOne (Max u v)) × (PlusOne (Max u v)))).
{ split; auto; apply Theorem74; Ens. }
clear H10 H15; elim H17; intros; apply Theorem158 in H17.
assert (P[En_Q u v x0] = P[f[[u, v]]]).
{ apply Theorem33 in H15; auto; clear H10 H12 H13 H14 H16 H17.
  apply Property_Value in H9; try apply H4;
  apply Property_ran in H9.
  apply Theorem154; Ens. }
rewrite H18 in H17; clear H11 H10 H15 H18.
apply Property_Value in H9;try apply H4;apply Property_ran in H9.
clear H12 H13 H14; apply H8 in H9; unfold R in H9;
apply AxiomII in H9.
destruct H9; apply Theorem4' in H2;
double H2; destruct H11 as [H11 _].
apply AxiomII in H11; destruct H11 as [_ H11], H11 as [H11 _].
apply AxiomII in H11; destruct H11 as [_ H11].
assert (ω ⊂ x0).
{ unfold Included; intros; unfold ω in H12; apply AxiomII in H12.
  destruct H12; double H13; destruct H14 as [H14 _], H2.
  apply AxiomII in H15; destruct H15 as [_ H15].
```

```
  add (Ordinal x0) H14; apply Theorem110 in H14.
  destruct H14 as [H14 | [H14 | H14]]; auto.
  - destruct H15; apply AxiomII; split; Ens.
    apply (Theorem132 z _); auto.
  - destruct H15; rewrite <- H14; apply AxiomII; Ens. }
apply H12 in H16; clear H12.
assert (P[f[[u, v]]] ∈ x0).
{ unfold LessEqual in H17; destruct H17; try rewrite H12; auto.
  destruct H11 as [_ H11]; apply H11 in H16; apply H16 in H12;
  auto. }
apply Lemma179'' in H12; destruct H2; auto. }
intros; generalize (classic (x0 = ω)); intros; destruct H13.
- rewrite H13 in H9; apply H12; auto.
- double H9; rewrite <- H6 in H9.
  unfold Cartesian in H14; apply AxiomII_P in H14;
  destruct H14, H15.
  clear H x; unfold Setminus in H2; apply Theorem4' in H2.
  destruct H2 as [H2 _]; double H2; unfold C in H2;
  apply AxiomII in H2.
  destruct H2 as [_ H2];
  unfold Cardinal_Number, Ordinal_Number in H2.
  destruct H2 as [H2 _]; apply AxiomII in H2;
  destruct H2 as [_ H2].
  clear H0; double H2; double H2;
  add (u ∈ x0) H2; add (v ∈ x0) H17.
  apply Theorem111 in H2; apply Theorem111 in H17.
  assert (Ordinal u /\ Ordinal v); auto.
  apply Theorem110 in H18;
  generalize (classic (Max u v ∈ ω)); intros.
  destruct H18 as [H18 | [H18 | H18]]; double H18.
  + apply Property_Max in H20; auto; rewrite H20 in *.
    clear H20; destruct H19.
    * apply H12; apply AxiomII_P; repeat split; auto.
      apply AxiomII in H19;destruct H19;apply AxiomII;split;Ens.
      apply Theorem132 in H18; auto.
    * assert (v ∈ (R ~ ω)).
      { unfold Setminus; apply Theorem4'; split;
        apply AxiomII; Ens. }
      apply Theorem174 in H20; clear H6 H7 H12 H14;
      destruct H4 as [H4 _].
      apply Property_Value in H9;auto; apply Property_ran in H9.
      apply H8 in H9; clear H8; assert (v ∈ R). apply AxiomII; Ens.
      apply Lemma123 in H6; AssE (PlusOne v); clear H6.
      assert(Ensemble((PlusOne v)×(PlusOne v)));
      try apply Theorem74;Ens.
```

```
double H10; apply Theorem33 in H8; auto.
add (En_Q u v x0 ⊂ (PlusOne v) × (PlusOne v)) H6; clear H10.
apply Theorem158 in H6; apply Theorem154 in H11; Ens.
rewrite H11 in H6; clear H8 H11; double H7;
apply Lemma179 in H7.
rewrite H7 in H6; clear H7; double H.
apply Theorem156 in H7; destruct H7 as [_ H7].
assert (P[v] ≺ P[x0]).
{ assert (Ordinal v /\ Ordinal x0); auto;
  apply Theorem118 in H10.
  assert (v ≼ x0); unfold LessEqual; try tauto;
  apply H10 in H11.
  assert (Ensemble x0 /\ v ⊂ x0); Ens;
  apply Theorem158 in H12.
  clear H10 H11; unfold LessEqual in H12; destruct H12; auto.
  apply Theorem154 in H10; Ens; apply Theorem146 in H10.
  unfold C in H; apply AxiomII in H; destruct H.
  apply H11 in H16; try contradiction; apply AxiomII; Ens. }
assert (P[(P[PlusOne v]) × (P[PlusOne v])] = P[PlusOne v]).
{ apply Property_PClass in H8.
  generalize (classic (P[(P[PlusOne v]) × (P[PlusOne v])] =
  P[PlusOne v])); intros; destruct H11; auto.
  assert (P[PlusOne v] ∈ \{ λ z, z ∈ (C ~ ω) /\
  P[z × z] <> z \}).
  { apply AxiomII; repeat split; Ens.
    unfold Setminus; apply Theorem4'; split; auto.
    apply AxiomII; split; Ens; intro; symmetry in H20.
    assert (v ∈ (PlusOne v)).
    { unfold PlusOne; apply Theorem4; right; unfold Singleton.
      apply AxiomII; split; Ens. }
    apply Theorem172 in H20; auto.
    - rewrite H20 in H14; generalize (Theorem101 v);
      contradiction.
    - assert (v ≼ (PlusOne v)); unfold LessEqual; auto.
      apply Theorem118 in H21; auto; clear H22; split; auto.
      assert (v ∈ R). apply AxiomII; Ens.
      apply Lemma123 in H22; apply AxiomII in H22;
      apply H22. }
  apply H1 in H12; destruct H12; unfold Rrelation, E.
  apply AxiomII_P; split; try apply Theorem49; Ens.
  rewrite H20, <- H7; auto. }
rewrite H11, H20 in H6; clear H11 H20; rewrite H7 in H10.
assert (P[f[[u, v]]] ∈ x0).
{ unfold LessEqual in H6; destruct H6; try rewrite H6; auto.
  destruct H0; apply H11 in H10; apply H10 in H6; auto. }
```

```
    unfold R in H9; apply AxiomII in H9; destruct H9.
    apply Lemma179'' in H11; auto.
+ apply Property_Max in H20; auto; rewrite Equal_Max in H20.
  rewrite H20 in *; clear H20; destruct H19.
  * apply H12; apply AxiomII_P; repeat split; auto.
    apply AxiomII in H19; destruct H19;
    apply AxiomII; split; Ens.
    apply Theorem132 in H18; auto.
  * assert (u ∈ (R ~ ω)).
    { unfold Setminus; apply Theorem4';
      split; apply AxiomII; Ens. }
    apply Theorem174 in H20; clear H6 H7 H12 H14;
    destruct H4 as [H4 _].
    apply Property_Value in H9;auto; apply Property_ran in H9.
    apply H8 in H9; clear H8; assert (u∈R).apply AxiomII;Ens.
    apply Lemma123 in H6; AssE (PlusOne u); clear H6.
    assert(Ensemble((PlusOne u)×(PlusOne u)));
    try apply Theorem74; Ens.
    double H10; apply Theorem33 in H8; auto.
    add (En_Q u v x0⊂(PlusOne u)×(PlusOne u)) H6;clear H10.
    apply Theorem158 in H6; apply Theorem154 in H11; Ens.
    rewrite H11 in H6; clear H8 H11; double H7;
    apply Lemma179 in H7.
    rewrite H7 in H6; clear H7; double H.
    apply Theorem156 in H7; destruct H7 as [_ H7].
    assert (P[u] ≺ P[x0]).
    { assert (Ordinal u /\ Ordinal x0); auto;
      apply Theorem118 in H10.
      assert (u ≼ x0); unfold LessEqual; try tauto;
      apply H10 in H11.
      assert (Ensemble x0 /\ u ⊂ x0); Ens;
      apply Theorem158 in H12.
      clear H10 H11;unfold LessEqual in H12;destruct H12;auto.
      apply Theorem154 in H10; Ens; apply Theorem146 in H10.
      unfold C in H; apply AxiomII in H; destruct H.
      apply H11 in H15;try contradiction;apply AxiomII;Ens. }
    assert (P[(P[PlusOne u]) × (P[PlusOne u])] = P[PlusOne u]).
    { apply Property_PClass in H8.
      generalize (classic (P[(P[PlusOne u]) × (P[PlusOne u])] =
      P[PlusOne u])); intros; destruct H11; auto.
      assert (P[PlusOne u] ∈ \{ λ z, z ∈ (C ~ ω) /\
      P[z × z] <> z \}).
      { apply AxiomII; repeat split; Ens.
        unfold Setminus; apply Theorem4'; split; auto.
        apply AxiomII; split; Ens; intro; symmetry in H20.
```

```
        assert (u ∈ (PlusOne u)).
        { unfold PlusOne; apply Theorem4; right; unfold Singleton.
          apply AxiomII; split; Ens. }
        apply Theorem172 in H20; auto.
        - rewrite H20 in H14; generalize (Theorem101 u);
          contradiction.
        - assert (u ≼ (PlusOne u)); unfold LessEqual; auto.
          apply Theorem118 in H21; auto; clear H22; split; auto.
          assert (u ∈ R). apply AxiomII; Ens.
          apply Lemma123 in H22; apply AxiomII in H22;
          apply H22. }
      apply H1 in H12; destruct H12; unfold Rrelation, E.
      apply AxiomII_P; split; try apply Theorem49; Ens.
      rewrite H20, <- H7; auto. }
    rewrite H11, H20 in H6; clear H11 H20; rewrite H7 in H10.
    assert (P[f[[u, v]]] ∈ x0).
    { unfold LessEqual in H6; destruct H6; try rewrite H6; auto.
      destruct H0; apply H11 in H10; apply H10 in H6; auto. }
    unfold R in H9; apply AxiomII in H9; destruct H9.
    apply Lemma179'' in H11; auto.
+ apply Property_Max in H20; auto; rewrite H18, H20 in *.
  clear H16 H18 H20; destruct H19.
  * apply H12; apply AxiomII_P; repeat split; auto.
  * assert (v ∈ (R ∼ ω)).
    { unfold Setminus; apply Theorem4';
      split; apply AxiomII; Ens. }
    apply Theorem174 in H18; clear H6 H7 H12 H14;
    destruct H4 as [H4 _].
    apply Property_Value in H9; auto; apply Property_ran in H9.
    apply H8 in H9; clear H8; assert (v∈R). apply AxiomII; Ens.
    apply Lemma123 in H6; AssE (PlusOne v); clear H6.
    assert(Ensemble((PlusOne v)×(PlusOne v)));
    try apply Theorem74;Ens.
    double H10; apply Theorem33 in H8; auto.
    add (En_Q v v x0⊂(PlusOne v)×(PlusOne v)) H6; clear H10.
    apply Theorem158 in H6; apply Theorem154 in H11; Ens.
    rewrite H11 in H6; clear H8 H11; double H7;
    apply Lemma179 in H7.
    rewrite H7 in H6; clear H7; double H.
    apply Theorem156 in H7; destruct H7 as [_ H7].
    assert (P[v] ≺ P[x0]).
    { assert (Ordinal v /\ Ordinal x0); auto;
      apply Theorem118 in H10.
      assert (v ≼ x0); unfold LessEqual; try tauto;
      apply H10 in H11.
```

```
          assert (Ensemble x0 /\ v ⊂ x0); Ens;
          apply Theorem158 in H12.
          clear H10 H11; unfold LessEqual in H12; destruct H12; auto.
          apply Theorem154 in H10; Ens; apply Theorem146 in H10.
          unfold C in H; apply AxiomII in H; destruct H.
          apply H11 in H15; try contradiction; apply AxiomII; Ens. }
        assert (P[(P[PlusOne v]) × (P[PlusOne v])] = P[PlusOne v]).
        { apply Property_PClass in H8.
          generalize (classic (P[(P[PlusOne v]) × (P[PlusOne v])] =
          P[PlusOne v])); intros; destruct H11; auto.
          assert (P[PlusOne v] ∈ \{ λ z, z ∈ (C ~ ω) /\
          P[z × z] <> z \}).
          { apply AxiomII; repeat split; Ens.
            unfold Setminus; apply Theorem4'; split; auto.
            apply AxiomII; split; Ens; intro; symmetry in H18.
            assert (v ∈ (PlusOne v)).
            { unfold PlusOne; apply Theorem4; right; unfold Singleton.
              apply AxiomII; split; Ens. }
            apply Theorem172 in H18; auto.
            - rewrite H18 in H14; generalize (Theorem101 v);
              contradiction.
            - assert (v ≼ (PlusOne v)); unfold LessEqual; auto.
              apply Theorem118 in H19; auto; clear H20; split; auto.
              assert (v ∈ R). apply AxiomII; Ens.
              apply Lemma123 in H20; apply AxiomII in H20;
              apply H20. }
          apply H1 in H12; destruct H12; unfold Rrelation, E.
          apply AxiomII_P; split; try apply Theorem49; Ens.
          rewrite H18, <- H7; auto. }
        rewrite H11, H18 in H6; clear H11 H18; rewrite H7 in H10.
        assert (P[f[[v, v]]] ∈ x0).
        { unfold LessEqual in H6; destruct H6; try rewrite H6; auto.
          destruct H0; apply H11 in H10; apply H10 in H6; auto. }
        unfold R in H9; apply AxiomII in H9; destruct H9.
        apply Lemma179'' in H11; auto. }
assert (P[x0 × x0] ≼ x0).
{ assert (P[dom(f)] = P[ran(f)]).
  { apply Theorem154; unfold Equivalent; Ens.
    assert (Ensemble x0 /\ Ensemble x0); auto.
    apply Theorem74 in H10; rewrite <- H6 in H10; split; auto.
    apply AxiomV; auto; apply H4. }
  assert (ran(f) ⊂ x0).
  { unfold Included; intros; unfold Range in H11;
    apply AxiomII in H11.
    destruct H11, H12; double H12;
```

```
          apply Property_dom in H13; double H13.
          apply Property_Value in H13;
          try apply H4; add ([x1, f[x1]] ∈ f) H12.
          apply H4 in H12; rewrite H12; clear H12 H13; rewrite H6 in H14.
          PP H14 a b; apply H9 in H12; auto. }
        add (ran( f) ⊂ x0) H0; apply Theorem158 in H0;
        unfold Setminus in H2.
        clear H11; apply Theorem4' in H2; destruct H2 as [H2 _].
        apply Theorem156 in H2; destruct H2 as [_ H2]; rewrite H2 in H0.
        rewrite <- H6, H10; auto. }
      unfold LessEqual in H10; destruct H10; try contradiction.
      assert (P[x0] ≼ P[x0 × x0]).
      { unfold Setminus in H2; apply Theorem4' in H2; destruct H2.
        unfold Complement in H11; apply AxiomII in H11;
        destruct H11 as [_ H11].
        generalize (classic (x0 = ∅)); intros; destruct H12.
        - rewrite H12 in H11; generalize (Theorem135 x); intros.
          destruct H13 as [H13 _]; contradiction.
        - apply Property_NotEmpty in H12; destruct H12 as [z H12].
          assert (P[x0] = P[x0 × [z]]).
          { apply Theorem154; try split; auto; try apply Lemma179'; Ens.
            apply Theorem74; split; try apply Theorem42; Ens. }
          rewrite H13; apply Theorem158;split;try apply Theorem74; auto.
          unfold Included; intros; PP H14 a b; apply AxiomII_P in H15.
          destruct H15,H16;unfold Singleton in H17;apply AxiomII in H17.
          destruct H17; apply AxiomII_P; repeat split; auto.
          rewrite H18; try apply Theorem19; Ens. }
      unfold LessEqual in H11; apply Theorem4' in H2;
      destruct H2 as [H2 _].
      double H2; apply Theorem156 in H2; destruct H2 as [_ H2], H11.
      + unfold C in H12; apply AxiomII in H12; destruct H12 as [_ H12].
        unfold Cardinal_Number, Ordinal_Number in H12;
        destruct H12 as [H12 _].
        apply AxiomII in H12; destruct H12 as [_ H12], H12 as [_ H12].
        unfold full in H12; apply H12 in H10; apply H10 in H11.
        rewrite H2 in H11; generalize (Theorem101 x0); intros;
        contradiction.
      + rewrite <- H11, H2 in H10; generalize (Theorem101 x0);
        contradiction.
Qed.

Hint Resolve Theorem179 : set.
```

定理 180 $(x \in C, y \in C, x \neq \emptyset \wedge y \neq \emptyset, x \notin \omega \vee y \notin \omega) \Longrightarrow P(x \times y) = \max[P(x), P(y)]$.

```
Lemma Lemma180 : ∀ x y, x × y ≈ y × x.
Proof.
  intros.
  unfold Equivalent; exists \{\ λ a b, a ∈ (x × y) /\ b ∈ [a]⁻¹ \}\.
  repeat split; intros; try (unfold Relation; intros; PP H a b; Ens).
  - destruct H; apply AxiomII_P in H; apply AxiomII_P in H0.
    destruct H, H0, H1, H2; unfold Singleton in H3, H4.
    PP H3 a b; PP H4 c d; clear H1 H2 H3 H4; apply AxiomII_P in H5.
    apply AxiomII_P in H6; destruct H5, H6; apply AxiomII in H2.
    apply AxiomII in H4; destruct H2, H4; clear H1 H2 H3 H4.
    apply Theorem49 in H; apply Theorem49 in H0; destruct H.
    destruct H0 as [_ H0]; assert (x0 ∈ 𝒰); try apply Theorem19; Ens.
    double H2; apply H5 in H2; apply H6 in H3; clear H5 H6.
    rewrite <- H3 in H2; clear H3; apply Theorem49 in H1; destruct H1.
    apply Theorem55 in H2; auto; destruct H2; rewrite H2, H4; auto.
  - destruct H; apply AxiomII_P in H; apply AxiomII_P in H0.
    destruct H, H0; apply AxiomII_P in H1; apply AxiomII_P in H2.
    destruct H1, H2; clear H H0 H1 H2; destruct H3, H4.
    PP H0 a b; PP H2 c d; apply AxiomII_P in H3; apply AxiomII_P in H4.
    destruct H3, H4; apply AxiomII in H5; apply AxiomII in H6.
    destruct H5, H6; rewrite H7 in H8; try apply Theorem19; Ens.
    apply H8; apply Theorem19; Ens.
  - apply AxiomI; split; intros.
    + apply AxiomII in H; destruct H, H0.
      apply AxiomII_P in H0; apply H0.
    + apply AxiomII; split; Ens; PP H a b; exists [b,a].
      apply AxiomII_P; assert (Ensemble ([a,b])); Ens.
      apply Theorem49 in H1; destruct H1.
      split; try (apply Theorem49; split; apply Theorem49; Ens).
      split; auto; unfold Inverse; apply AxiomII_P.
      split; try apply Theorem49; auto; unfold Singleton.
      apply AxiomII; split; try apply Theorem49; Ens.
  - apply AxiomI; split; intros.
    + apply AxiomII in H; destruct H, H0.
      apply AxiomII_P in H0; destruct H0, H1; PP H1 a b; clear H1.
      apply AxiomII_P in H3; destruct H3, H3; unfold Inverse in H2.
      PP H2 c d; clear H2; apply AxiomII_P in H5; destruct H5.
      clear H0 H2; unfold Singleton in H5; apply AxiomII in H5.
      destruct H5; apply Theorem49 in H0.
      assert ([a, b] ∈ 𝒰); try apply Theorem19; Ens; apply H2 in H5.
      apply Theorem55 in H5; auto; clear H0 H2; destruct H5.
      rewrite H0, H2; unfold Cartesian; apply AxiomII_P.
      repeat split; try apply Theorem49; Ens.
    + unfold Range; apply AxiomII; split; Ens; PP H a b.
      apply AxiomII_P in H0; destruct H0, H1; double H0; exists [b,a].
```

```
    apply Theorem49 in H3; destruct H3; apply AxiomII_P.
    repeat split; try (apply Theorem49; split; apply Theorem49; Ens).
    * unfold Cartesian; apply AxiomII_P; repeat split; auto.
      apply Theorem49; split; auto.
    * unfold Inverse, Singleton; apply AxiomII_P; split; auto.
      apply AxiomII; split; try apply Theorem49; Ens.
Qed.

Lemma Lemma180' : ∀ x y,
  x ∈ C -> y ∈ C -> x ∉ ω -> x ≠ ∅ -> y ≠ ∅ ->
  P[x × y] = Max P[x] P[y].
Proof.
  intros.
  assert (x ∈ (C ~ ω)).
  { unfold Setminus; apply Theorem4'; split; auto.
    unfold Complement; apply AxiomII; split; Ens. }
  apply Theorem179 in H4; double H; double H0; unfold C in H5, H6.
  apply AxiomII in H5; apply AxiomII in H6; destruct H5, H6.
  unfold Cardinal_Number, Ordinal_Number in H7, H8; destruct H7, H8.
  clear H5 H6 H9 H10; apply AxiomII in H7; apply AxiomII in H8.
  destruct H7, H8; assert (Ordinal x /\ Ordinal y); auto.
  apply Theorem110 in H9; destruct H9 as [H9 | [H9 | H9]].
  - assert (Ensemble (y × y) /\ (x × y) ⊂ (y × y)).
    { split; unfold Included; intros.
      - apply Theorem74; split; auto.
      - unfold Ordinal,full in H8;destruct H8 as [_H8];apply H8 in H9.
        PP H10 a b; clear H10; apply AxiomII_P in H11; unfold Cartesian.
        apply AxiomII_P; destruct H11, H11; repeat split; auto. }
    apply Theorem158 in H10;apply Property_NotEmpty in H2;destruct H2.
    assert (y ≈ ([x0] × y)).
    { apply Theorem147 with (y:= y × [x0]); try apply Lemma179'; Ens.
      apply Lemma180. }
    assert (Ensemble (x × y) /\ ([x0] × y) ⊂ (x × y)).
    { split; try apply Theorem74; Ens; unfold Included; intros.
      PP H12 a b; apply AxiomII_P in H13; destruct H13, H14.
      apply AxiomII_P; repeat split; auto; unfold Singleton in H14.
      apply AxiomII in H14; destruct H14; rewrite H16; auto.
      apply Theorem19; Ens. }
    assert (Ensemble y /\ Ensemble ([x0] × y)).
    { split; auto; apply Theorem74; split; auto; apply Theorem42; Ens.}
    apply Theorem158 in H12; apply Theorem154 in H13; apply H13 in H11.
    rewrite <- H11 in H12; clear H11 H13.
    assert (y ∈ (C ~ ω)).
    { unfold Setminus; apply AxiomII; repeat split; auto.
      unfold Complement; apply AxiomII; split; auto; intro.
```

```
      unfold ω in H11; apply AxiomII in H11; destruct H11 as [_ H11].
      apply Theorem132 in H9; auto; destruct H1; unfold ω.
      apply AxiomII; split; auto. }
    apply Theorem179 in H11; rewrite H11 in H10; clear H5 H7 H11.
    apply Theorem156 in H; apply Theorem156 in H0; destruct H, H0.
    rewrite H5, H7 in *; apply Property_Max in H9; auto; rewrite H9.
    apply Lemma174; auto.
  - assert (Ensemble (x × x) /\ (x × y) ⊂ (x × x)).
    { split; unfold Included; intros; try (apply Theorem74; Ens).
      unfold Ordinal, full in H6; destruct H6 as [_ H6]; apply H6 in H9.
      PP H10 a b; clear H10; apply AxiomII_P in H11; unfold Cartesian.
      apply AxiomII_P; destruct H11, H11; repeat split; auto. }
    apply Theorem158 in H10; rewrite H4 in H10; clear H4.
    apply Property_NotEmpty in H3; destruct H3.
    assert (x ≈ (x × [x0])); try apply Lemma179'; Ens.
    assert (Ensemble (x × y) /\ (x × [x0]) ⊂ (x × y)).
    { split; try apply Theorem74; Ens; unfold Included; intros.
      PP H11 a b; apply AxiomII_P in H12; destruct H12, H13.
      apply AxiomII_P; repeat split; auto; unfold Singleton in H13.
      apply AxiomII in H14; destruct H14; rewrite H15; auto.
      apply Theorem19; Ens. }
    assert (Ensemble x /\ Ensemble (x × [x0])).
    { split; auto; apply Theorem74; split; auto; apply Theorem42; Ens.}
    apply Theorem158 in H11; apply Theorem154 in H12; apply H12 in H4.
    rewrite <- H4 in H11; clear H4 H5 H7 H12; apply Theorem156 in H.
    apply Theorem156 in H0; destruct H, H0; rewrite H4, H5 in *.
    apply Property_Max in H9; auto; rewrite Equal_Max in H9.
    rewrite H9; apply Lemma174; auto.
  - rewrite <- H9 in *; clear H0 H1 H2 H3 H5 H6 H7 H8 H9.
    apply Theorem156 in H; destruct H; rewrite H0; assert (x=x); auto.
    apply Property_Max in H1; rewrite H1; auto.
Qed.

Theorem Theorem180 : ∀ x y,
  x ∈ C -> y ∈ C -> x ∉ ω \/ y ∉ ω -> x ≠ ∅ -> y ≠ ∅ ->
  P[x × y] = Max P[x] P[y].
Proof.
  intros; destruct H1.
  - apply Lemma180'; auto.
  - assert (x × y ≈ y × x); try apply Lemma180.
    assert (Ensemble (x × y) /\ Ensemble (y × x)).
    { split; apply Theorem74; split; Ens. }
    apply Theorem154 in H5; apply H5 in H4; clear H5.
    rewrite H4, Equal_Max; apply Lemma180'; auto.
Qed.
```

```
Hint Resolve Theorem180 : set.
```

这里定理 180 的叙述与文献 [41] 有所不同, 增加了条件 “$x \neq 0 \wedge y \neq 0$”, 这是在我们证明过程中发现的, 事实上, 可以证明如下的定理.

定理 180′　$\exists x, y, x \in C, y \in C, x \notin \omega, P(x \times y) \neq \max[P(x), P(y)]$.

```
Lemma Theorem180_Not :
  ∃ x y, x ∈ C /\ y ∈ C /\ x ∉ ω /\ P[x × y] <> Max P[x] P[y].
Proof.
  exists ω, ∅; generalize (Theorem135 ∅); intros.
  destruct H as [H _]; double H; apply Theorem164 in H0.
  repeat split; try apply Theorem165; try apply Theorem101; auto.
  generalize Theorem165; intros; apply Theorem156 in H0.
  apply Theorem156 in H1; destruct H0 as [_ H0], H1 as [_ H1].
  assert (ω × ∅ = ∅).
  { apply AxiomI; split; intros.
    - PP H2 a b; apply AxiomII_P in H3; destruct H3, H4.
      generalize (Theorem16 b); intros; contradiction.
    - generalize (Theorem16 z); intros; contradiction. }
  rewrite H2, H0, H1; clear H0 H1 H2; double H; unfold ω in H0.
  apply AxiomII in H0; destruct H0 as [_ H0], H0 as [H0 _].
  generalize Property_ω; intros; double H.
  apply Property_Max in H2; auto; rewrite Equal_Max in H2; rewrite H2.
  intro; rewrite H3 in H; generalize (Theorem101 ω); contradiction.
Qed.
```

$C \sim \omega$ 的元被称为 “**无限基数**”, 或者 “**超穷基数**”.

定理 181　*存在唯一的 <-< 保序函数以 R 为定义域, 并以 $C \sim \omega$ 为值域.*

```
Theorem Theorem181 : ∃ f, Order_Pr f E E /\ dom(f) = R /\
  ran(f) = C ∼ ω.
Proof.
  generalize Theorem113; intros; destruct H; apply Theorem107 in H.
  assert ((C ∼ ω) ⊂ R).
  { unfold Included, Setminus; intros;
    apply Theorem4' in H1; destruct H1.
    unfold C in H1; apply AxiomII in H1; unfold Cardinal_Number in H1.
    destruct H1, H3; unfold Ordinal_Number in H3; auto. }
  apply Lemma97 with (r:= E) in H1; auto;
  add (WellOrdered E (C ∼ ω)) H.
  clear H1; apply Theorem99 in H; destruct H as [f H], H, H1; exists f.
  destruct H1, H3, H4, H5; split; auto; destruct H2; split; auto.
  - rewrite <- H2 in H0; clear H2 H5.
```

```
    apply Theorem96 in H4; destruct H4 as [H2 _], H2.
    generalize (classic (ran(f) = C ∼ ω)); intros; destruct H5; auto.
    assert (Ensemble ran(f)).
    { unfold Section in H6; destruct H6, H7 as [_ H7].
      assert (ran(f) ⊊ C ∼ ω); unfold ProperIncluded; auto.
      apply Property_ProperIncluded' in H8; destruct H8, H8.
      assert (ran(f) ⊂ x).
      { unfold Included; intros; double H10; apply H6 in H11.
        assert (x ∈ (C ∼ ω) /\ z ∈ (C ∼ ω)); auto.
        unfold WellOrdered in H3; destruct H3 as [H3 _].
        apply H3 in H12; destruct H12 as [H12 | [H12 | H12]].
        - destruct H9; apply H7 with (v:= z); auto.
        - unfold Rrelation, E in H12; apply AxiomII_P in H12; apply H12.
        - rewrite H12 in H9; contradiction. }
      apply Theorem33 in H10; Ens. }
    rewrite Lemma96 in H0; rewrite Lemma96' in H7.
    apply AxiomV in H7; auto; contradiction.
  - clear H3 H4 H0 H6.
    generalize (classic (dom(f) = R)); intros; destruct H0; auto.
    assert (∼ Ensemble ran(f)).
    { rewrite H2; intro; generalize Theorem162, Theorem165; intros.
      add (Ensemble ω) H3; Ens; apply AxiomIV in H3; clear H6.
      assert (C ∼ ω ∪ ω = C).
      { apply AxiomI; unfold Setminus; split; intros.
        - apply Theorem4 in H6; destruct H6.
          + apply Theorem4' in H6; apply H6.
          + generalize Theorem164; intros; apply H7 in H6; auto.
        - generalize (classic (z ∈ ω)); intros; apply Theorem4.
          destruct H7; try tauto; left; apply Theorem4'.
          split; auto; unfold Complement; apply AxiomII; Ens. }
      rewrite H6 in H3; contradiction. }
    assert (Ensemble dom(f)); clear H2.
    { intros; generalize Theorem113; intros;
      destruct H2 as [H2 _]; double H5.
      apply Theorem114 in H5;
      assert (Ordinal dom(f) /\ Ordinal R); auto.
      apply Theorem110 in H6; destruct H6 as [H6|[H6|H6]];
      try tauto; Ens.
      apply H4 in H6; generalize(Theorem101 R); intros; contradiction.}
    apply AxiomV in H4; auto; contradiction.
Qed.

Theorem Theorem181' : forall f g,
  Order_Pr f E E -> Order_Pr g E E -> dom(f) = R -> dom(g) = R ->
  ran(f) = C ∼ ω -> ran(g) = C ∼ ω -> f = g.
```

```
Proof.
  intros.
  assert (Order_Pr f E E /\ Order_Pr g E E); auto.
  generalize Theorem113; intros;
  destruct H6 as [H6 _]; apply Theorem107 in H6.
  assert ((C ~ ω) ⊂ R).
  { unfold Included, Setminus; intros;
    apply Theorem4' in H7; destruct H7.
    unfold C in H7; apply AxiomII in H7; unfold Cardinal_Number in H7.
    destruct H7, H9; unfold Ordinal_Number in H9; auto. }
  apply Lemma97 with (r:= E) in H7; auto.
  assert (Section dom(f) E R /\ Section dom(g) E R).
  { rewrite H1, H2; unfold Section, Included.
    split; try (repeat split; try apply H6;
    intros; auto; try apply H8). }
  assert (Section ran(f) E (C ~ ω) /\ Section ran(g) E (C ~ ω)).
  { rewrite H3, H4; unfold Section, Included.
    split; try (repeat split; try apply H7;
    intros; auto; try apply H9). }
  apply (Theorem97 f g E E R (C ~ ω)) in H5; auto; clear H6 H7 H8 H9.
  unfold Order_Pr in H, H0; destruct H, H0, H5.
  - apply Theorem27; split; auto; unfold Included; intros.
    rewrite Theorem70; rewrite Theorem70 in H8; auto; PP H8 a b.
    double H9; rewrite <- Theorem70 in H9; auto; apply AxiomII_P in H10.
    destruct H10; apply AxiomII_P; split; auto; rewrite H11 in *.
    assert ([a,f[a]] ∈ f).
    { apply Property_Value; auto; apply Property_dom in H9.
      rewrite H2, <- H1 in H9; auto. }
    apply H5 in H12; eapply H0; eauto.
  - apply Theorem27; split; auto; unfold Included; intros.
    rewrite Theorem70; rewrite Theorem70 in H8; auto; PP H8 a b.
    double H9; rewrite <- Theorem70 in H9; auto; apply AxiomII_P in H10.
    destruct H10; apply AxiomII_P; split; auto; rewrite H11 in *.
    assert ([a,g[a]] ∈ g).
    { apply Property_Value; auto; apply Property_dom in H9.
      rewrite H1, <- H2 in H9; auto. }
    apply H5 in H12; eapply H; eauto.
Qed.

Hint Resolve Theorem181 Theorem181' : set.

End A11.

Export A11.
```

这个唯一的 <-< 保序函数的存在性由前面的定理所保证, 它通常用 $\aleph$ 来表示. 于是 $\aleph(0)$ (或者 $\aleph_0$) 为 ω. 紧接的下一个基数 $\aleph_1$ 也用 Ω 来表示. 因此它是第一个不可数序数. 由于 $P(2^{\aleph_0}) > \aleph_0$ 推得 $P(2^{\aleph_0}) \geqslant \aleph_1$. 这两个基数的相等是极有吸引力的猜测, 它被称为 "**连续统假设**". 广义连续统假设是这样叙述的: 如果 x 是一个序数, 则 $P(2^{\aleph_x}) = \aleph_{x+1}$. 此假设既不能被证明也不能被否定. Gödel 曾证明如下的一致性定理 [21, 22].

定理(Gödel 一致性定理)[21, 22]　*如果在连续统假设的基础上产生了一个矛盾, 则矛盾也可以在不假定连续统假设的情况下被找到. 对广义连续统假设和选择公理也是一样的.*

1966 年, Cohen 证明了如下的相容性与独立性定理 [6, 16].

定理(Cohen 相容性与独立性定理)[6, 16]　*连续统假设 (以及广义连续统假设和选择公理) 与一般的集合论公理是相容并独立的.*

我们当然可以继续对 Gödel 一致性定理 [21, 22] 和 Cohen 相容性与独立性定理 [6, 16] 进行形式化证明实现, 国外也有一些相关的尝试工作, 但我们不准备展开了, 因为这需要引入一些新的概念和方法, 而就一般的数学基础来说, 上面已经完成的 Morse-Kelley 公理化系统已是足够了.

第 4 章　选择公理及其等价命题的机器证明

选择公理是集合论里有关映射存在性的一条公理, 最早于 1904 年由 Zermelo 提出, 并用于对良序定理的证明. 选择公理在现代数学中有很重要的作用, 与许多深刻的数学结论有着十分密切的联系. 没有选择公理, 甚至无法确定两个集合能否比较元素的多少、非空集的积是否非空、线性空间是否一定有一组基、环是否一定有极大理想等等. 选择公理有多个等价定理, 包括 Tukey 引理、Hausdorff 极大原则、Zermelo 假定、Zorn 引理、良序定理等. 拓扑学中重要的 Tychonoff 乘积定理[40, 41, 84] 即选择公理较为深刻的一个应用.

如图 4.1 所示, 我们从选择公理出发, 依次证明 Tukey 引理、Hausdorff 极大原则、极大原则、Zermelo 假定、Zorn 引理和良序定理, 再将选择公理视为一条定理, 分别由 Tukey 引理、Zermelo 假定及良序定理证明选择公理, 完成整个循环策略的证明, 从而说明上述各命题与选择公理等价. 这些命题的人工证明过程是标准的, 散见于拓扑学或集合论的相关专著或教材中, 例如, 可参见文献 [41, 84].

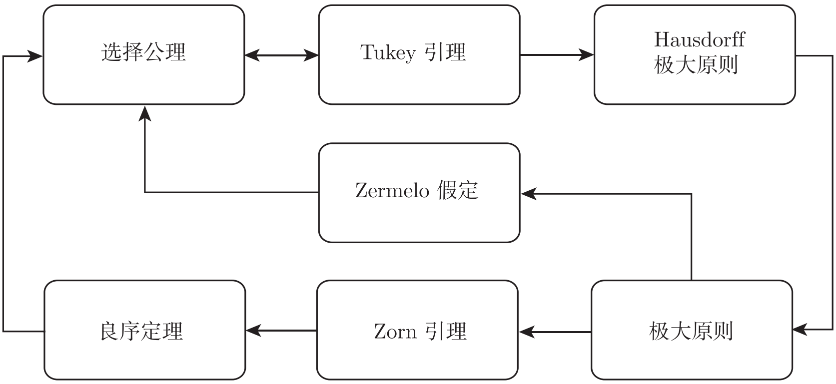

图 4.1　选择公理及其等价命题的联系图

4.1　基 本 定 义

在证明选择公理与相关定理等价性之前需要引入一些基本定义, 这些基本定义将在后面的证明中反复使用.

为了本章应用部分的独立性和完整性, 我们在这里重新定义了选择函数并在之

后章节中重新给出选择公理的描述. 本章中的定义虽和 "公理化集合论" 形式化系统中的略有区别, 但二者实质相同.

```
Require Export A_11.
(** 证明选择公理等价性所必要的定义 **)
Module BasicDefinition.
```

定义 (选择函数) 设 X 为一个集合, ε 为一个选择函数的充要条件为 ε 是一个函数, ε 的定义域为 X 的所有非空子集构成的集合, 即 $2^X - 0$, ε 的值域为 X, 且对任意 ε 的定义域中的 A, $\varepsilon(A) \in A$.

```
Definition Choice_Function ε X : Prop :=
  Function ε /\ ran( ε ) ⊂ X /\ dom( ε ) = pow( X ) ~ [ ∅ ] /\
  (∀ A, A ∈ dom( ε ) -> ε[ A ] ∈ A).

Hint Unfold Choice_Function : Axiom_of_Choice.
```

定义 (极大 (极小) 成员) 设 $\mathcal{F}$ 是一个集族, F 是 $\mathcal{F}$ 的一个成员. 如果 $\mathcal{F}$ 中没有任何成员以 F 为真子集 (如果 $\mathcal{F}$ 中没有任何成员为 F 的真子集), 则称 F 为 $\mathcal{F}$ 的一个极大 (极小) 成员.

```
Definition MaxMember F 𝓕 : Prop :=
  𝓕 ≠ ∅ -> F ∈ 𝓕 /\ (∀ x, x ∈ 𝓕 -> ~ F ⊊ x).
Definition MinMember F 𝓕 : Prop :=
  𝓕 ≠ ∅ -> F∈ 𝓕 /\ (∀ x, x∈ 𝓕 -> ~ x ⊊ F).

Hint Unfold MaxMember MinMember : Axiom_of_Choice.
```

定义 (套) 设 $\mathcal{F}$ 是一个集族, 如果对于任意 $A, B \in \mathcal{F}$ 有 $A \subset B$ 或者 $B \subset A$ 则称 $\mathcal{F}$ 为一个套.

```
Definition Nest 𝓕 : Prop := ∀ A B, A ∈𝓕 /\ B ∈ 𝓕 ->
  A ⊂ B \/ B ⊂ A.

Hint Unfold Nest : Axiom_of_Choice.
```

定义 (有限特征集) 设 $\mathcal{F}$ 是一个集族. 如果 F 是 $\mathcal{F}$ 的一个成员当且仅当 F 的每一个有限子集都是 $\mathcal{F}$ 的成员, 则称 $\mathcal{F}$ 是一个具有有限特征的集族.

```
Definition FiniteSet 𝓕 : Prop :=
  Ensemble 𝓕 /\ (∀ F, F ∈ 𝓕 -> (∀ A, A ⊂ F /\ Finite A -> A ∈ 𝓕))
  /\ (∀ F, Ensemble F /\ (∀ A, A ⊂ F /\ Finite A -> A ∈ 𝓕) -> F ∈ 𝓕).

Hint Unfold FiniteSet : Axiom_of_Choice.
```

定理 (有限特征集性质) 如果 $\mathcal{F}$ 是一个非空的具有有限特征的集合, 则

(1) $\mathcal{F}$ 中每一个成员的任何一个子集都是 $\mathcal{F}$ 的成员;

(2) $\mathcal{F}$ 中任何一个套的并都是 $\mathcal{F}$ 的成员.

```
Proposition Property_FinSet : ∀ ℱ: Class,
  FiniteSet ℱ /\ ℱ ≠ ∅ -> (∀ A B, A ∈ ℱ /\ B ⊂ A -> B ∈ ℱ)
  /\ (∀ φ, φ ⊂ ℱ /\ Nest φ -> (⋃φ) ∈ ℱ).
Proof.
  intros; destruct H.
  unfold FiniteSet in H; destruct H; split; intros.
  - destruct H2; apply H1; intros; split.
    + apply Theorem33 in H3; Ens.
    + intros; destruct H4; apply H1 with (z:=z) in H2; auto.
      split; try (apply Theorem28 with (y:=B); split); auto.
  - destruct H2; apply H1.
    split; try (apply AxiomVI; apply Theorem33 in H2); auto.
    intro A; intros; destruct H4; unfold Finite in H5.
    generalize (classic (φ = ∅)); intros; destruct H6.
    + rewrite H6 in *; clear H6; rewrite Theorem24' in H4.
      add (∅ ⊂ A) H4; try apply (Theorem26 A); apply Theorem27 in H4.
      rewrite H4 in *; clear H4; apply Property_NotEmpty in H0.
      destruct H0; generalize (Theorem26 x); intros.
      apply H1 with (z:= ∅) in H0; auto.
    + assert (Ensemble A).
      { apply Theorem33 in H2; auto; apply AxiomVI in H2.
        apply Theorem33 in H4; auto. }
      double H7; apply Theorem153 in H8; apply Theorem146 in H8.
      assert (∀ D, D ∈ \{ λ D, D ≈ P[A] /\ D ⊂ ⋃ φ \} ->
             ∃ B, B ∈ φ /\ D ⊂ B).
      { apply Mathematical_Induction with (n:= P[A]); auto; intros.
        - apply AxiomII in H9; destruct H9, H10.
          generalize (classic (D = ∅)); intros; destruct H12.
          + rewrite H12 in *; apply Property_NotEmpty in H6;
            destruct H6.
            exists x; split; auto; apply Theorem26.
          + unfold Equivalent in H10;
            destruct H10 as [g H10], H10, H13, H10.
            apply Property_NotEmpty in H12;
            destruct H12; rewrite <- H13 in H12.
            apply Property_Value in H12; auto;
            apply Property_ran in H12.
            rewrite H14 in H12; generalize (Theorem16 g[x]);
            contradiction.
        - clear H H0 H2 H4 H5 H6 H7 H8 A.
          destruct H9; apply AxiomII in H10; destruct H10, H4.
```

```
unfold Equivalent in H4; destruct H4 as [g H4], H4, H4, H6.
assert (k ∈ (PlusOne k)).
{ unfold PlusOne; apply Theorem4; right; unfold Singleton.
  apply AxiomII; split; Ens. }
rewrite <- H8 in H9; unfold Range in H9; apply AxiomII in H9.
destruct H9, H10; double H10; apply Property_dom in H11.
rewrite H6 in H11.
assert ((D ~ [x]) ∈ \{ λ D0, D0 ≈ k /\ D0 ⊂ ⋃ φ \}).
{ apply AxiomII; repeat split.
  - unfold Setminus; apply Theorem33 with (x:= D); auto.
    unfold Included; intros; apply Theorem4' in H12;
    apply H12.
  - clear H0 H1; unfold Equivalent; exists (g | (D ~ [x])).
    repeat split; unfold Relation; intros.
    + unfold Restriction in H0; apply Theorem4' in H0.
      destruct H0; PP H1 a b; Ens.
    + destruct H0; apply Theorem4' in H0;
      apply Theorem4' in H1.
      destruct H0 as [H0 _], H1 as [H1 _].
      apply H4 with (x:= x0); auto.
    + PP H0 a b; Ens.
    + destruct H0; apply AxiomII_P in H0;
      apply AxiomII_P in H1.
      destruct H0, H1; apply Theorem4' in H12;
      apply Theorem4' in H13.
      destruct H12 as [H12 _], H13 as [H13 _];
      apply H7 with (x:=x0).
      split; apply AxiomII_P; Ens.
    + apply AxiomI; split; intros.
      * unfold Domain in H0; apply AxiomII in H0;
        destruct H0, H1.
        unfold Restriction in H1; apply Theorem4' in H1;
        destruct H1.
        unfold Cartesian in H12; apply AxiomII_P in H12;
        apply H12.
      * unfold Domain; apply AxiomII; split; Ens.
        double H0; unfold Setminus in H1;
        apply Theorem4' in H1.
        destruct H1 as [H1 _]; rewrite <- H6 in H1.
        apply Property_Value in H1; auto; exists g[z].
        unfold Restriction; apply Theorem4'; split; auto.
        unfold Cartesian; apply AxiomII_P; repeat split; Ens.
        AssE [z,g[z]]; apply Theorem49 in H12; destruct H12.
        apply Theorem19; auto.
    + apply AxiomI; split; intros.
```

```
        * unfold Range in H0; apply AxiomII in H0;
          destruct H0, H1.
          unfold Restriction in H1; apply Theorem4' in H1.
          destruct H1; unfold Cartesian in H12;
          apply AxiomII_P in H12.
          destruct H12, H13 as [H13 _]; unfold Setminus in H13.
          apply Theorem4' in H13; destruct H13; double H1.
          apply Property_ran in H15; rewrite H8 in H15.
          apply Theorem4 in H15; destruct H15; auto; clear H0.
          unfold Singleton in H15; apply AxiomII in H15;
          destruct H15.
          rewrite H15 in H1; try apply Theorem19; Ens;
          clear H0 H12 H15.
          assert ([k,x] ∈ g⁻¹ /\ [k,x0] ∈ g⁻¹).
          { split; apply AxiomII_P; split;
          try apply Theorem49; Ens. }
          apply H7 in H0; rewrite <- H0 in H14;
          apply AxiomII in H14.
          destruct H14, H14; apply AxiomII; auto.
        * unfold Range; apply AxiomII; split; Ens.
          assert (z ∈ ran(g)).
          { rewrite H8; apply Theorem4; tauto. }
          apply AxiomII in H1; destruct H1, H12; exists x0.
          unfold Restriction; apply Theorem4'; split; auto.
          unfold Cartesian; apply AxiomII_P; split; Ens.
          split; try apply Theorem19; auto; unfold Setminus.
          double H12; apply Property_dom in H13;
          rewrite H6 in H13.
          apply Theorem4'; split; auto; unfold Complement.
          apply AxiomII; split; Ens; intro; apply AxiomII in H14.
          destruct H14; rewrite H15 in H12;
          try apply Theorem19; Ens.
          add ([x, z] ∈ g) H10; apply H4 in H10;
          rewrite H10 in H0.
          generalize (Theorem101 z); intros; contradiction.
    - unfold Included, Setminus; intros; apply Theorem4' in H12.
      destruct H12; apply H5 in H12; auto. }
  apply H0 in H12; clear H0; destruct H12 as [B H12];
  apply H5 in H11.
  clear H4 H6 H7 H8 H9 H10; apply AxiomII in H11; destruct H11.
  destruct H4 as [C H4], H4, H12; assert (B∈ φ /\ C∈ φ); auto.
  unfold Nest in H3; apply H3 in H9; destruct H9.
  + add (B ⊂ C) H8; apply Theorem28 in H8; clear H9.
    exists C; split; auto; unfold Included; intros.
    generalize (classic (z = x)); intros; destruct H10.
```

```
          * rewrite H10; auto.
          * apply H8; unfold Setminus; apply Theorem4'; split; auto.
            unfold Complement; apply AxiomII; split; Ens; intro.
            destruct H10; apply AxiomII in H11; apply H11.
            apply Theorem19; auto.
        + apply H9 in H4; clear H9; exists B; split; auto;
          unfold Included.
          intros; generalize (classic (z = x)); intros; destruct H10.
          * rewrite H10; auto.
          * apply H8; unfold Setminus; apply Theorem4'; split; auto.
            unfold Complement; apply AxiomII; split; Ens; intro.
            destruct H10; apply AxiomII in H11; apply H11.
            apply Theorem19; auto. }
    assert (A ∈ \{ λ D0, D0 ≈ P[A] /\ D0 ⊂ ⋃ φ \}).
    { apply AxiomII; repeat split; auto. }
    apply H9 in H10; clear H9; destruct H10 as [B H10], H10.
    apply H2 in H9; apply H1 with (z:= A) in H9; auto.
Qed.
```

定义 (偏序、偏序集) 设 X 是一个集合, $\preceq$ 是 X 上的一个偏序, $(X,\preceq)$ 是一个偏序集当且仅当 $\preceq$ 是 X 上的一个关系, 且满足:

(1) 自反性: 对任意 $x\in A$, 有 $x\preceq x$;

(2) 反对称性: 对任意 $x,y\in A$, 若 $x\preceq y$, 且 $y\preceq x$, 则 $x=y$;

(3) 传递性: 对任意 $x,y,z\in A$, 若 $x\preceq y$, 且 $y\preceq z$, 则 $x\preceq z$.

```
Definition Reflexivity le X := ∀ a, a ∈ X -> Rrelation a le a.

Definition Antisymmetry le X :=
  ∀ x y, x ∈ X /\ y ∈ X ->
  (Rrelation x le y /\ Rrelation y le x -> x = y).

Definition Transitivity r x : Prop :=
  ∀ u v w, (u ∈ x /\ v ∈ x /\ w ∈ x) /\
  (Rrelation u r v /\ Rrelation v r w) -> Rrelation u r w.

Definition PartialOrder le X : Prop :=
  Ensemble X /\ (Relation le /\ le ⊂ X × X) /\
  Reflexivity le X /\ Antisymmetry le X /\ Transitivity le X.

Definition PartialOrderSet X le := PartialOrder le X.

Hint Unfold PartialOrder PartialOrderSet: Axiom_of_Choice.
```

定义 (上 (下) 界) 设 $(X,\preceq)$ 是一个偏序集, 设 $A\subset X$. 如果点 $x\in X$ 使得

对于每一个 $a \in A$ 都有 $a \preceq x$ (或 $x \preceq a$), 则称 x 是 A 的一个上界 (下界).

```
Definition BoundU x A X le : Prop :=
  PartialOrder le X /\ X ≠ ∅ ->
  x ∈ X /\ A ⊂ X /\ (∀ a, a ∈ A -> Rrelation a le x).

Definition BoundD x A X le : Prop :=
  PartialOrder le X /\ X ≠ ∅ ->
  x ∈ X /\ A ⊂ X /\ (∀ a, a ∈ A -> Rrelation x le a).

Hint Unfold BoundU BoundD : Axiom_of_Choice.
```

定义 (极大 (极小) 元素)　设 $(X, \preceq)$ 是一个偏序集, $x \in X$ 称为一个极大 (极小) 元素, 如果对于任何 $y \in X$, 都不存在 $x \preceq y$ ($y \preceq x$) 且 $y \neq x$ 的情况.

```
Definition MaxElement x X le : Prop :=
  X ≠ ∅ -> x ∈ X /\ (∀ y, y ∈ X -> ~ (Rrelation x le y /\ x ≠ y)).

Definition MinElement x X le : Prop :=
  X ≠ ∅ -> x ∈ X /\ (∀ y, y ∈ X -> ~ (Rrelation y le x /\ x ≠ y)).

Hint Unfold MaxElement MinElement : Axiom_of_Choice.
```

定义 (全序、全序集)　设 $(X, \preceq)$ 是一个偏序集, 如果它满足条件: 对于任何 $x, y \in X$, 若 $x \neq y$, 则或者 $x \preceq y$, 或者 $y \preceq x$. 便称 $\preceq$ 是 X 上的一个全序, 也称 $(X, \preceq)$ 为一个全序.

```
Definition TotalOrder le X :=
  PartialOrder le X /\ (∀ x y, x ∈ X /\ y ∈ X ->
  (x ≠ y -> Rrelation x le y \/ Rrelation y le x)).

Definition TotalOrderSet X le := TotalOrder le X.

Hint Unfold TotalOrder TotalOrderSet : Axiom_of_Choice.
```

定义 (链)　设 $(X, \preceq)$ 是一个偏序集, A 是 X 的一个非空子集. 如果集合 A 对于这个偏序而言是一个全序集, 便称 A 为 X 中的一个链.

```
Definition Chain A X le : Prop :=
  PartialOrder le X ->(A ⊂ X /\ A ≠ ∅) /\ TotalOrder (le ∩ (A × A))A.

Hint Unfold Chain : Axiom_of_Choice.
```

定义 (良序、良序集)　设 $(X, \preceq)$ 是一个全序集, 如果在 X 的每一个非空子集

A 中有一个极小元素, 即存在 $a \in A$ 使得对于任何 $x \in A$ 都有 $a \preceq x$, 则称 $\preceq$ 是 X 上的一个良序, 也称 $(X, \preceq)$ 为一个良序集.

```
Definition WellOrder le X :=
  TotalOrder le X /\ (∀ A, A ⊂ X /\ A ≠ ∅ -> ∃ z, MinElement z A le).

Definition WellOrderSet X le := WellOrder le X.

Hint Unfold WellOrder WellOrderSet : Axiom_of_Choice.
```

定义 (截)　设 $(X, \preceq)$ 是一个良序集. $Y \subset X$ 称为一个截, 如果它满足条件: 对任意的 $u \in X, v \in Y$, 如果 $u \preceq v$, 则 $u \in Y$.

```
Definition Initial_Segment Y X le := Y ⊂ X /\ WellOrder le X /\
  (∀ u v, (u ∈ X /\ v ∈ Y /\ Rrelation u le v ) -> u ∈ Y).

Hint Unfold Initial_Segment : Axiom_of_Choice.

End BasicDefinition.

Export BasicDefinition.
```

4.2 Tukey 引理

首先我们根据选择公理证明 Tukey 引理. 为了完整起见, 给出通过选择函数描述的选择公理如下.

公理 (选择公理)　任何一个集合都有一个选择函数.

```
Require Export Basic_Definitions.

Module Type Axiom_Choice.

Axiom Choice_Axiom : ∀ X, Ensemble X ->
  ∃ε, Choice_Function ε X.

Hint Resolve Choice_Axiom : Axiom_of_Choice.

End Axiom_Choice.
```

Tukey 引理是由 Teichmüller (1939 年) 与 Tukey (1940 年)[37, 70] 提出的, 是与选择公理相关的重要定理. 下面我们根据选择公理证明 Tukey 引理.

```
Module Type Tukey_Lemma.

Declare Module AC : Axiom_Choice.
```

定理 (Tukey 引理) 非空的具有有限特征的集合中必有极大成员.

假设 $\mathcal{F}$ 为一个非空的具有有限特征的集族, 令 $X = \bigcup\mathcal{F}$. 根据选择公理, X 有选择函数 $\varepsilon : \tilde{X} \to X$ 使得对于任意 $A \in \tilde{X}$, $\varepsilon(A) \in A$, 其中 $\tilde{X}$ 为 X 的所有非空子集构成的集族.

对每一个 $F \in \mathcal{F}$, 记

$$\hat{F} = \{x : x \in X \wedge F \cup \{x\} \in \mathcal{F}\}$$

对 $\hat{F}$ 的形式化定义记作 En_$\hat{F}$, 具体如下：

```
Definition En_F̂ F 𝓕 : Class :=
  \{ λ x, x ∈ (⋃𝓕) /\ (F ∪ [x]) ∈ 𝓕 \}.
```

$\hat{F}$ 显然是 X 的子集. 根据 $\hat{F}$ 的定义可知, $F \subset \hat{F}$ 或者 $F = \hat{F}$. 定义一个函数 $\chi : \mathcal{F} \to \mathcal{F}$ 使得对任意的 $F \in \mathcal{F}$

$$\chi(F) = \begin{cases} F \cup \{\varepsilon(F' - F)\}, & F' - F \neq 0 \\ F, & F' - F = 0 \end{cases}$$

显然, $\chi(F)$ 或者等于 F 或者比 F 多一点, 并且 F 是 $\mathcal{F}$ 的极大成员当且仅当 $\chi(F) = F$.

对函数 χ 的 Coq 形式化分为三部分完成. 第一部分首先定义一个判定函数 eq_dec, 该函数对任意类型为 Type 的变量都满足; 第二部分将特定类型 Class 作为函数 eq_dec 中变元的类型, 从而得到新函数 eq_dec1; 第三部分在函数 eq_dec1 基础上定义函数 Function_χ. 在 Coq 中通过模式匹配定义, 具体实现如下：

```
Definition eq_dec (A : Type) := ∀ x y: A, {x = y} + {x <> y}.
Parameter eq_dec1 : eq_dec Class.
Definition Function_χ (F 𝓕 ε: Class) : Class :=
  match eq_dec1 ((En_F̂ F 𝓕) ~ F) ∅ with
  | left _ => F
  | right _ => F ∪ [ε[(En_F̂ F 𝓕)~F]]
  end.
```

通过 χ 函数的定义, Tukey 引理的证明目标可转化为判定是否存在一个 $\mathcal{F}$ 中的元 F, 使得 $\chi(F) = F$. 为此引入 t-子族定义.

定义 (t-子族) $\mathcal{T}$ 是 $\mathcal{F}$ 的 t-子族当且仅当 $\mathcal{T}$ 是 $\mathcal{F}$ 的子族并且满足条件:

(1) $0 \in \mathcal{T}$;

(2) 若 $F \in \mathcal{T}$, 则 $\chi(F) \in \mathcal{T}$;

(3) 若 $\mathcal{C}$ 为 $\mathcal{T}$ 中的套, 则 $\bigcup\mathcal{C} \in \mathcal{T}$.

```
Definition tSubset 𝒯 ℱ ε : Prop :=
  𝒯 ⊂ ℱ /\ ∅ ∈𝒯 /\ (∀ F, F ∈ 𝒯 -> (Function_χ F ℱ ε) ∈𝒯) /\
  (∀ φ, φ ⊂ 𝒯 /\ Nest φ -> (⋃φ) ∈ 𝒯).
```

令 $\mathcal{F}_0$ 为 $\mathcal{F}$ 中所有 t-子族的交, 可证 $\mathcal{F}_0$ 是一个套. 为此, 对于每一个 $C \in \mathcal{F}_0$, 令

$$\mu(C) = \{A : A \in \mathcal{F}_0 \wedge (A \subset C \vee C \subset A)\}$$

通过 $\mathcal{F}_0$ 构建集合 $\mathcal{F}_1$:

$$\mathcal{F}_1 = \{C : C \in \mathcal{F}_0 \wedge \mu(C) = \mathcal{F}_0\}$$

对于每一个 $D \in \mathcal{F}_1$, 令

$$\nu(D) = \{A : A \in \mathcal{F}_1 \wedge (A \subset D \vee \chi(D) \subset A)\}$$

上述公式的 Coq 形式化如下:

```
Definition En_ℱ₀ ℱ ε := ⋂ \{ λ 𝒯, tSubset 𝒯 ℱ ε \}.

Definition En_μ C ℱ ε := \{ λ A, A ∈ (En_ℱ₀ ℱ ε) /\ (A ⊂ C \/
  C ⊂ A) \}.

Definition En_ℱ₁ ℱ ε : Class :=
  \{ λ C, C ∈ (En_ℱ₀ ℱ ε) /\ (En_μ C ℱ ε) = (En_ℱ₀ ℱ ε) \}.

Definition En_ν D ℱ ε : Class :=
  \{ λ A, A ∈ (En_ℱ₀ ℱ ε) /\ (A ⊂ D \/ (Function_χ D ℱ ε) ⊂ A) \}.
```

另外, 容易证明关于 $\mathcal{F}_0$ 和函数 χ 的如下性质, 它们将在后面的定理证明中反复使用. 形式化代码及证明如下:

```
Lemma Property_χ : ∀ ε F ℱ,
  Choice_Function ε (⋃ℱ) -> F ∈ ℱ -> F ⊂ (Function_χ F ℱ ε).
Proof.
  intros.
  generalize (classic ((En_F̂ F ℱ) ~ F = ∅)); intros; destruct H1.
```

```
  - unfold Function_χ; destruct (beq (En_F̂ F 𝓕 ∼ F) ∅); try tauto.
    unfold Included; intros; auto.
  - unfold Function_χ; destruct (beq (En_F̂ F 𝓕 ∼ F) ∅); try tauto.
    unfold Included; intros; apply Theorem4; tauto.
Qed.

Lemma Ens_F'F : ∀ 𝓕 F, Ensemble (⋃ 𝓕) ->
  Ensemble (En_F̂ F 𝓕 ∼ F).
Proof.
  intros.
  assert (En_F̂ F 𝓕 ∼ F ⊂ ⋃ 𝓕).
  { unfold Included; intros.
    unfold Setminus in H0; apply Theorem4' in H0; destruct H0.
    unfold En_F̂ in H0; apply AxiomII in H0; apply H0. }
  apply Theorem33 in H0; auto.
Qed.

Lemma Property_𝓕0 : ∀ 𝓕 ε,
  FiniteSet 𝓕 /\ 𝓕 ≠ ∅ -> Choice_Function ε (⋃𝓕) ->
  tSubset (En_𝓕0 𝓕 ε) 𝓕 ε
  /\ (∀ 𝓣, 𝓣 ⊂ 𝓕 /\ tSubset 𝓣 𝓕 ε -> (En_𝓕0 𝓕 ε) ⊂ 𝓣).
Proof.
  intros; double H.
  apply Property_FinSet in H; unfold FiniteSet in H1; destruct H1, H1.
  apply Property_NotEmpty in H2. destruct H2 as [F H2]; split.
  - assert (tSubset 𝓕 𝓕 ε).
    { unfold tSubset; repeat split; try apply H; intros.
      - unfold Included; intros; auto.
      - generalize (Theorem26 F); intros.
        apply H with (A:= F); split; auto.
      - generalize (classic((En_F̂ F0 𝓕) ∼ F0 = ∅)); intros;
        destruct H5.
        + unfold Function_χ;destruct (beq (En_F̂ F0 𝓕 ∼ F0)∅); tauto.
        + double H5; unfold Function_χ.
          destruct (beq (En_F̂ F0 𝓕 ∼ F0) ∅); try tauto.
          unfold Choice_Function in H0; destruct H0, H7, H8.
          assert ((En_F̂ F0 𝓕 ∼ F0) ∈ dom( ε)).
          { rewrite H8; unfold Setminus at 2; apply Theorem4'; split.
            - unfold PowerSet; apply AxiomII; apply AxiomVI in H1.
              split; try (apply Ens_F'F); auto.
              unfold Included; intros; unfold Setminus in H10.
              apply Theorem4' in H10; destruct H10.
              unfold En_F̂ in H10; apply AxiomII in H10; apply H10.
```

```
          - unfold Complement; apply AxiomII; double H1.
            apply AxiomVI in H10; split; try (apply Ens_F'F); auto.
            intro; unfold Singleton in H11; apply AxiomII in H11.
            destruct H11; rewrite H12 in H6; try contradiction.
            apply Theorem19; generalize (Theorem26 𝓕); intros.
            apply Theorem33 in H13; auto. }
         apply H9 in H10; unfold Setminus in H10.
         apply Theorem4' in H10; destruct H10.
         unfold En_F̂ at 2 in H10; apply AxiomII in H10; apply H10. }
  assert ((En_𝓕₀ 𝓕 ε) ⊂ 𝓕).
  { unfold En_𝓕₀; unfold Included; intros.
    unfold Element_I in H5; apply AxiomII in H5.
    apply H5; apply AxiomII; split; auto. }
  unfold tSubset; repeat split; auto.
  + unfold En_𝓕₀; apply AxiomII; split; intros.
    * generalize (Theorem26 𝓕); intros; apply Theorem33 in H6; auto.
    * apply AxiomII in H6; destruct H6;
      unfold tSubset in H7; apply H7.
  + intros; double H6; unfold Included in H5.
    apply H5 in H7; unfold tSubset in H4; apply H4 in H7.
    unfold En_𝓕₀; apply AxiomII; split; intros; Ens.
    apply AxiomII in H8; destruct H8.
    double H9; unfold tSubset in H9; apply H9.
    unfold En_𝓕₀ in H6; apply AxiomII in H6; destruct H6.
    apply H11; apply AxiomII; split; auto.
  + intros; unfold tSubset in H4; destruct H6; double H6.
    add ((En_𝓕₀ 𝓕 ε) ⊂ 𝓕) H6; apply Theorem28 in H6.
    add (Nest φ) H6; apply H4 in H6; unfold En_𝓕₀; apply AxiomII.
    split; Ens; intros; apply AxiomII in H9; destruct H9.
    unfold tSubset in H10; apply H10; split; auto.
    assert ((En_𝓕₀ 𝓕 ε) ⊂ y).
    { unfold Included; intros; unfold En_𝓕₀ in H11.
      unfold Element_I; apply AxiomII in H11.
      destruct H11; apply H12; apply AxiomII; split; auto. }
    add ((En_𝓕₀ 𝓕 ε) ⊂ y) H8; apply Theorem28 in H8; auto.
 - intros; destruct H4; unfold Included; intros.
   unfold En_𝓕₀ in H6; unfold Element_I in H6.
   apply AxiomII in H6; destruct H6; apply H7.
   apply AxiomII; apply Theorem33 in H4; auto.
Qed.

Lemma FF' : ∀ 𝓕 ε F,
  FiniteSet 𝓕 /\ 𝓕 ≠ ∅ -> Choice_Function ε (⋃𝓕) -> F ∈ 𝓕 ->
  (En_F̂ F 𝓕)~F ≠ ∅ -> F = F ∪ [ε[(En_F̂ F 𝓕)~F]] -> False.
```

```
Proof.
  intros.
  unfold Choice_Function in H0; assert (F ∼ F = ∅).
  { unfold ∅; apply AxiomI; split; intros.
    - unfold Setminus in H4; apply Theorem4' in H4; destruct H4.
      apply AxiomII in H5; destruct H5; contradiction.
    - apply AxiomII in H4; destruct H4; contradiction. }
  assert (F ∼ F ≠ ∅); try contradiction.
  { rewrite H3 at 1; apply Property_NotEmpty;
    exists (ε [En_F̂ F 𝓕 ∼ F ]).
    assert ((En_F̂ F 𝓕 ∼ F) ∈ dom( ε)).
    { destruct H0, H5, H6; rewrite H6; assert (En_F̂ F 𝓕 ∼ F ⊂ ⋃ 𝓕).
      { unfold Included; intros; apply Theorem4' in H8; destruct H8.
        unfold En_F̂ in H8; apply AxiomII in H8; apply H8. }
      assert (Ensemble (En_F̂ F 𝓕 ∼ F)).
      { apply Theorem33 in H8; auto; destruct H.
        unfold FiniteSet in H; destruct H; apply AxiomVI; auto. }
      unfold Setminus at 2; apply Theorem4'; split.
      - unfold PowerSet; apply AxiomII; split; auto.
      - unfold Complement; apply AxiomII; split; auto.
        unfold NotIn; intro; unfold Singleton in H10.
        apply AxiomII in H10; destruct H10; assert (∅ ∈ 𝓤).
        { apply Theorem19; generalize (Theorem26 (⋃ 𝓕)); intros.
          unfold FiniteSet in H; destruct H.
          apply Theorem33 in H12; auto; apply AxiomVI; apply H. }
        apply H11 in H12; contradiction. }
    apply H0 in H5; unfold Setminus in H5.
    apply AxiomII in H5; destruct H5, H6; unfold Setminus.
    apply Theorem4'; split; auto; apply Theorem4; right.
    unfold Singleton; apply AxiomII; split; auto. }
Qed.

Lemma Property_F' : ∀ F 𝓕, F ∈ 𝓕 -> F ⊂ (En_F̂ F 𝓕).
Proof.
  unfold En_F̂, Included; intros.
  apply AxiomII; repeat split; Ens.
  - unfold Element_U; apply AxiomII; split; Ens.
  - assert (F ∪ [z] = F).
    { apply AxiomI; split; intros; try (apply Theorem4; tauto).
      apply Theorem4 in H1; destruct H1; auto.
      unfold Singleton in H1; apply AxiomII in H1.
      destruct H1; rewrite H2; auto; apply Theorem19; Ens. }
    rewrite H1; auto.
Qed.
```

下面将分五步完成 Tukey 引理的证明. 前三步证明了 $\mathcal{F}_0$ 是一个套. 第四步证明了当 $F = \bigcup \mathcal{F}_0$ 时, $\chi(F) = F$. 最后, 根据前四步证明的引理在第五步中证明 Tukey 引理.

第一步通过之前的定义和性质证明引理 T1, 其描述如下.

引理 T1 如果 D 是集合 $\mathcal{F}_1$ 中的一个元素, 则 $\nu(D)$ 是非空有限特征集 $\mathcal{F}$ 的一个 t-子族.

在形式化过程中当调用 t-子族的形式化定义时, 必须声明参数 ε. 因此在引理的条件中加入 ε 是 $\bigcup \mathcal{F}$ 的选择函数.

```
Lemma LemmaT1 : ∀ 𝓕 ε,
  FiniteSet 𝓕 /\ 𝓕 ≠ ∅ -> Choice_Function ε (⋃𝓕) ->
  (∀ D, D ∈ (En_𝓕₁ 𝓕 ε) -> tSubset (En_ν D 𝓕 ε) 𝓕 ε).
Proof.
  intros.
  apply (Property_𝓕₀ _ ε) in H; auto; destruct H.
  assert ((En_ν D 𝓕 ε) ⊂ 𝓕).
  { unfold En_ν; unfold Included; intros.
    apply AxiomII in H3; destruct H3, H4.
    unfold tSubset in H; destruct H.
    unfold Included in H; apply H in H4; auto. }
  unfold tSubset; repeat split; auto.
  - unfold En_ν; apply AxiomII.
    unfold tSubset in H; destruct H, H4.
    repeat split; Ens; left; apply Theorem26.
  - intro A; intros.
    double H4; unfold Included in H3; apply H3 in H4.
    unfold En_ν in H5; unfold En_ν.
    apply AxiomII; apply AxiomII in H5; destruct H5, H6.
    double H6; unfold tSubset in H; apply H in H8.
    repeat split; Ens; destruct H7.
    + apply Property_ProperIncluded in H7. destruct H7.
      * left; generalize (classic ((Function_χ A 𝓕 ε) ⊂ D)); intros.
        destruct H9; auto; unfold En_𝓕₁ in H1; apply AxiomII in H1.
        destruct H1, H10; rewrite <- H11 in H8.
        unfold En_μ in H8; apply AxiomII in H8; destruct H8, H12.
        apply Property_ProperIncluded'' in H13; auto.
        double H7; apply Property_ProperIncluded' in H7.
        double H13; apply Property_ProperIncluded' in H13.
        destruct H7, H13, H7, H13.
        generalize (classic (x = x0)); intros; destruct H18.
        -- rewrite H18 in H7; contradiction.
        -- unfold ProperIncluded in H15; destruct H15.
           unfold Included in H15; apply H15 in H7.
```

```
          assert (x0 ∉ A).
          { unfold NotIn; intro; unfold ProperIncluded in H14.
            destruct H14; unfold Included in H14.
            apply H14 in H20; contradiction. }
          generalize(classic((En_F̂ A ℱ)~A=∅));intros;destruct H21.
          ++ unfold Function_χ in H13; unfold NotIn in H20.
             destruct (beq (En_F̂ A ℱ ~ A) ∅); tauto.
          ++ unfold Function_χ in H7, H8, H13.
             destruct (beq (En_F̂ A ℱ ~ A) ∅); try tauto.
             apply Theorem4 in H7; apply Theorem4 in H13.
             destruct H7, H13; try contradiction.
             unfold Singleton in H13; apply AxiomII in H13.
             unfold Singleton in H7; apply AxiomII in H7;
             destruct H13, H7.
             apply AxiomIV' in H8; destruct H8.
             apply Theorem42' in H24; apply Theorem19 in H24.
             double H24; apply H22 in H24; apply H23 in H25.
             rewrite <- H24 in H25; contradiction.
      * right; rewrite H7.
        unfold Included; intros; auto.
    + apply (Property_χ ε _ _) in H4; auto.
      add (A ⊂ Function_χ A ℱ ε) H7; apply Theorem28 in H7; auto.
  - intro ϑ; intros; destruct H4.
    unfold En_ν; apply AxiomII.
    assert ((⋃ ϑ) ∈ (En_ℱ₀ ℱ ε)).
    { unfold tSubset in H; apply H; split; auto.
      red; intros; unfold Included in H4.
      apply H4 in H6; unfold En_ν in H6.
      apply AxiomII in H6; apply H6. }
    repeat split; Ens.
    generalize (classic (∀ B, B ∈ ϑ -> B ⊂ D)).
    intros; destruct H7.
    + left; unfold Included; intros.
      unfold Element_U in H8; apply AxiomII in H8.
      destruct H8, H9, H9; apply H7 in H10.
      unfold Included in H10; apply H10 in H9; auto.
    + apply not_all_ex_not in H7; destruct H7.
      apply imply_to_and in H7; destruct H7.
      double H7; unfold Included in H4; apply H4 in H7.
      unfold En_ν in H7; apply AxiomII in H7.
      destruct H7, H10, H11; try contradiction.
      right; unfold Included; intros.
      unfold Element_U; apply AxiomII; split; Ens.
Qed.
```

引理 T1 证明了 $\nu(D)$ 是 $\mathcal{F}$ 的一个 t- 子族. 第二步通过引理 T1 证明 $\mathcal{F}_1$ 满足 t- 子族定义中的条件 2. 具体描述如下:

引理 T2 如果 D 是集合 $\mathcal{F}_1$ 中的一个元, 则 $\chi(D)$ 同样是 $\mathcal{F}_1$ 中的一个元.

```
Lemma LemmaT2 : ∀ 𝓕 ε,
  FiniteSet 𝓕 /\ 𝓕 ≠ ∅ -> Choice_Function ε (⋃𝓕) ->
  (∀ D, D ∈ (En_𝓕1 𝓕 ε) -> (Function_χ D 𝓕 ε) ∈ (En_𝓕1 𝓕 ε)).
Proof.
  intros; double H1.
  unfold En_𝓕1 in H2; apply AxiomII in H2.
  destruct H2, H3; double H3; unfold En_𝓕0 in H5.
  double H; apply (Property_𝓕0 _ ε) in H6; auto.
  destruct H6; unfold tSubset in H6.
  double H3; apply H6 in H8; double H8.
  unfold En_𝓕0 in H9; destruct H6.
  unfold Included in H6; apply H6 in H3.
  apply (Property_χ ε _ _) in H3; auto.
  assert ((En_ν D 𝓕 ε) ⊂ (En_μ (Function_χ D 𝓕 ε) 𝓕 ε)).
  { unfold En_ν, En_μ, Included; intros.
    apply AxiomII in H11; apply AxiomII; destruct H11, H12.
    repeat split; auto; destruct H13; auto. }
  assert ((En_μ (Function_χ D 𝓕 ε) 𝓕 ε) ⊂ (En_𝓕0 𝓕 ε)).
  { unfold En_μ, Included; intros.
    apply AxiomII in H12; apply H12. }
  apply (LemmaT1 𝓕 ε) in H1; auto.
  unfold FiniteSet in H; destruct H, H.
  assert ((En_ν D 𝓕 ε) ⊂ 𝓕 /\ tSubset (En_ν D 𝓕 ε) 𝓕 ε).
  { split; auto; unfold En_ν, Included; intros.
    apply AxiomII in H15; destruct H15, H16.
    apply H6 in H16; auto. }
  apply H7 in H15;
  add ((En_ν D 𝓕 ε) ⊂ (En_μ (Function_χ D 𝓕 ε) 𝓕 ε)) H15.
  apply Theorem28 in H15.
  add ((En_𝓕0 𝓕 ε) ⊂ (En_μ (Function_χ D 𝓕 ε) 𝓕 ε)) H12.
  apply Theorem27 in H12; auto.
  unfold En_𝓕1; apply AxiomII; repeat split; Ens.
Qed.
```

接下来第三步通过引理 T2 证明 $\mathcal{F}_0$ 是一个套. 具体描述和其 Coq 形式化如下:

引理 T3 如果 $\mathcal{F}$ 为一个非空的具有有限特征的集族, ε 是 $\bigcup\mathcal{F}$ 的选择函数, 则 $\mathcal{F}_0$ 是一个套.

```
Lemma LemmaT3 : ∀ 𝓕 ε,
```

```
  FiniteSet ℱ /\ ℱ ≠ ∅ -> Choice_Function ε (⋃ℱ) -> Nest (En_ℱ0 ℱ ε).
Proof.
  intros; double H.
  apply (Property_ℱ0 _ ε) in H1; auto; destruct H1.
  assert ((En_ℱ1 ℱ ε) ⊂ (En_ℱ0 ℱ ε) /\ Nest (En_ℱ1 ℱ ε)).
  { assert ((En_ℱ1 ℱ ε) ⊂ (En_ℱ0 ℱ ε)).
    { unfold Included; intros.
      unfold En_ℱ1 in H3; apply AxiomII in H3; apply H3. }
    split; auto; unfold tSubset in H1.
    add ((En_ℱ0 ℱ ε) ⊂ ℱ) H3; try apply H1.
    apply Theorem28 in H3; unfold FiniteSet in H; destruct H, H.
    apply Theorem33 with (z:=(En_ℱ1 ℱ ε)) in H; auto.
    unfold Nest; intros; unfold En_ℱ1 in H6; destruct H6.
    apply AxiomII in H6; apply AxiomII in H7.
    destruct H6, H7, H8, H9; rewrite <- H11 in H8.
    unfold En_μ in H8; apply AxiomII in H8; apply H8. }
  destruct H3.
  assert ((En_ℱ1 ℱ ε) ⊂ ℱ /\ tSubset (En_ℱ1 ℱ ε) ℱ ε).
  { unfold tSubset in H1.
    add ((En_ℱ0 ℱ ε) ⊂ ℱ) H3; try apply H1.
    apply Theorem28 in H3; split; auto.
    unfold tSubset; repeat split; auto; intros.
    - unfold En_ℱ1; apply AxiomII.
      destruct H1, H5; repeat split; Ens.
      apply AxiomI; split; intros.
      + unfold En_μ; apply AxiomII in H7; apply H7.
      + unfold En_μ; apply AxiomII; repeat split; Ens.
        right; apply Theorem26.
    - apply (LemmaT2 _ ε); auto.
    - unfold En_ℱ1; apply AxiomII.
      assert ((⋃ φ) ∈ (En_ℱ0 ℱ ε)).
      { destruct H5; assert (φ ⊂ (En_ℱ0 ℱ ε)).
        { unfold Included; intros.
          unfold Included in H5; apply H5 in H7.
          unfold En_ℱ1; apply AxiomII in H7; apply H7. }
        add (Nest φ) H7; apply H1 in H7; auto. }
      repeat split; Ens.
      apply AxiomI; split; intros.
      + unfold En_μ in H7; apply AxiomII in H7; apply H7.
      + unfold En_μ; apply AxiomII; repeat split; Ens.
        generalize (classic (∀ B, B ∈ φ -> B ⊂ z)).
        intros; destruct H8.
        * right; unfold Included; intros.
          unfold Element_U in H9; apply AxiomII in H9.
          destruct H9, H10, H10; apply H8 in H11.
```

```
        unfold Included in H11; apply H11; auto.
      * apply not_all_ex_not in H8; destruct H8.
        apply imply_to_and in H8; destruct H5, H8.
        double H8; unfold Included in H5; apply H5 in H8.
        unfold En_𝓕₁ in H8; apply AxiomII in H8; destruct H8, H12.
        rewrite <- H13 in H7; unfold En_μ in H7; apply AxiomII in H7.
        destruct H7, H14; destruct H15; try contradiction.
        left; unfold Included; intros.
        unfold Element_U; apply AxiomII; split; Ens. }
  apply H2 in H5; add ((En_𝓕₁ 𝓕 ε) ⊂ (En_𝓕₀ 𝓕 ε)) H5.
  apply Theorem27 in H5; auto; rewrite H5; auto.
Qed.
```

在证明了 $\mathcal{F}_0$ 是一个套之后, 第四步证明存在某一元 $F \in \mathcal{F}$ 使得 $\chi(F) = F$, 具体引理描述及 Coq 形式化描述如下所示:

引理 T4 $\bigcup\mathcal{F}_0$ 是 $\mathcal{F}$ 中的一个元并且 $\chi(\bigcup\mathcal{F}_0) = \bigcup\mathcal{F}_0$.

```
Lemma LemmaT4 : ∀ 𝓕 ε,
  FiniteSet 𝓕 /\ 𝓕 ≠ ∅ -> Choice_Function ε (⋃𝓕) ->
  (⋃ En_𝓕₀ 𝓕 ε) ∈ 𝓕 /\
  (Function_χ (⋃ (En_𝓕₀ 𝓕 ε)) 𝓕 ε) = ⋃ (En_𝓕₀ 𝓕 ε).
Proof.
  intros; double H.
  apply (Property_𝓕₀ _ ε) in H1; auto.
  destruct H1; unfold tSubset in H1.
  assert ((En_𝓕₀ 𝓕 ε) ⊂ (En_𝓕₀ 𝓕 ε) /\ Nest (En_𝓕₀ 𝓕 ε)).
  { split; try unfold Included; auto.
    apply (LemmaT3 _ ε) in H; auto. }
  apply H1 in H3; split.
  - destruct H1; unfold Included in H1; apply H1 in H3; auto.
  - unfold En_𝓕₀ at 2 in H3; destruct H1.
    unfold Included in H1; double H3; apply H1 in H5.
    apply (Property_χ ε _ _) in H5; auto.
    assert ((Function_χ (⋃ (En_𝓕₀ 𝓕 ε)) 𝓕 ε) ⊂ ⋃ (En_𝓕₀ 𝓕 ε)).
    { apply H4 in H3; unfold Included; intros.
      unfold Element_U; apply AxiomII; split; Ens. }
    apply Theorem27; auto.
Qed.
```

最后证明的第五步通过引理 T4 证明 Tukey 引理, 也就是证明非空的具有有限特征的集族必有极大成员. 具体证明过程如下:

```
Theorem Tukey : ∀ (𝓕: Class),
  FiniteSet 𝓕 /\ 𝓕 ≠ ∅ -> ∃ x, MaxMember x 𝓕.
Proof.
  intros; double H.
```

```
  unfold FiniteSet in H0; destruct H0, H0.
  assert (Ensemble (⋃𝓕)). { apply AxiomVI in H0; auto. }
  apply AC.Choice_Axiom in H3; destruct H3 as [ε H3].
  assert (∃ F, F∈𝓕 /\ (En_F̂ F 𝓕) ∼ F = ∅).
  { exists (⋃(En_𝓕0 𝓕 ε)); double H3.
    apply (LemmaT4 _ ε) in H4; auto; destruct H4; split; auto.
    generalize (classic(En_F̂(⋃ En_𝓕0 𝓕 ε)𝓕∼(⋃ En_𝓕0 𝓕 ε)=∅)).
    intros; destruct H6; auto.
    assert ((Function_χ (⋃ (En_𝓕0 𝓕 ε)) 𝓕 ε) = (⋃ En_𝓕0 𝓕 ε) ∪
    [ε [En_F̂ (⋃ En_𝓕0 𝓕 ε) 𝓕 ∼ (⋃ En_𝓕0 𝓕 ε)]]).
    { unfold Function_χ;
      destruct (beq (En_F̂ (⋃ En_𝓕0 𝓕 ε) 𝓕 ∼
      (⋃ En_𝓕0 𝓕 ε)) ∅); tauto. }
    rewrite H5 in H7; apply FF' in H7; auto; inversion H7. }
  destruct H4 as [F H4]; destruct H4; exists F.
  apply -> Property_∅ in H5; try (apply Property_F'; auto).
  unfold MaxMember; intro; clear H6; repeat split; auto;
  intros F' H6; intro.
  double H7; rewrite <- H5 in H8; apply Property_ProperIncluded' in H8.
  destruct H8 as [z H8], H8; assert (F' ⊂ (En_F̂ F 𝓕)).
  { unfold En_F̂, Included; intros; apply AxiomII; repeat split; Ens.
    - unfold Element_U; apply AxiomII; split; Ens.
    - assert ((F ∪ [z0]) ⊂ F').
      { unfold ProperIncluded in H7; destruct H7.
        unfold Included in H7; unfold Included; intros.
        apply Theorem4 in H12;destruct H12;try (apply H7 in H12;auto).
        unfold Singleton in H12; apply AxiomII in H12.
        destruct H12; rewrite H13; auto; apply Theorem19; Ens. }
      apply Property_FinSet in H; apply H with (A:= F'); split; auto. }
  unfold Included in H10; apply H10 in H8; contradiction.
Qed.

Hint Resolve Tukey : Axiom_of_Choice.

End Tukey_Lemma.
```

4.3 Hausdorff 极大原则

定理 (Hausdorff 极大原则) *设 $\mathcal{A}$ 是一个集合, $\mathcal{N}$ 是 $\mathcal{A}$ 中的一个套, 则 $\mathcal{A}$ 中存在一个极大的套 μ 包含着 $\mathcal{N}$.*

为证 Hausdorff 极大原则, 我们需要如下的两个引理. 其中引理 H1 可视为

Tukey 引理的一个推论.

引理 H1 若 $\mathcal{F}$ 是一个具有有限特征的集族, 并且 $A \in \mathcal{F}$, 则 $\mathcal{F}$ 中包含 A 一个极大成员.

引理 H2 设 $\mathcal{A}$ 是一个集族, 则 $\mathcal{A}$ 中所有的套构成的集族是具有有限特征的.

上述引理和 Hausdorff 极大原则的 Coq 描述及证明代码如下:

```
Require Export Tukey_Lemma.

Module Type Husdorff_Principle.

Declare Module Tu : Tukey_Lemma.

(* LemmaH1 *)

Definition En_f1 f A := \{ λ F, F∈f /\ (F ∪ A) ∈ f \}.

Lemma LemmaH1_1 : ∀ (f A: Class),
  FiniteSet f -> A∈f -> FiniteSet (En_f1 f A).
Proof.
  intros.
  assert (f ≠ ∅). { apply Property_NotEmpty; try exists A; auto. }
  double H; add (f ≠ ∅) H2; apply Property_FinSet in H2; destruct H2.
  unfold FiniteSet in H; destruct H; unfold FiniteSet; repeat split.
  - unfold En_f1; assert (\{ λ F,F∈f /\ (F∪A)∈f \} ⊂ f).
    { unfold Included; intros; apply AxiomII in H5; apply H5. }
    apply Theorem33 in H5; auto.
  - intro B; intro; intro B1; intros; unfold En_f1 in H5.
    apply AxiomII in H5; destruct H5, H7; unfold En_f1; apply AxiomII.
    elim H6; intros; apply H4 in H6; auto; repeat split; Ens.
    assert ((B1 ∪ A) ⊂ (B ∪ A)).
    { unfold Included; intros; apply Theorem4 in H11.
      apply Theorem4; destruct H11; try tauto.
      unfold Included in H9; apply H9 in H11; auto. }
    add (B1∪A ⊂ B∪A) H8; apply H2 in H8; auto.
  - intro B; intros; destruct H5.
    unfold En_f1; apply AxiomII; repeat split; auto.
    + apply H4; split; auto; intros; apply H6 in H7.
      unfold En_f1 in H7; apply AxiomII in H7; apply H7.
    + apply H4; split; try (apply AxiomIV; split; Ens).
      intro A1; intros; destruct H7.
      assert ((B∩A1) ⊂ B /\ Finite (B∩A1)).
      { split.
        - unfold Included; intros; apply Theorem4' in H9; apply H9.
```

```
      - rewrite Theorem6'; apply Finite_Included with (A:=A1); auto.
        unfold Included; intros; apply Theorem4' in H9; apply H9. }
    apply H6 in H9; unfold En_f1 in H9; apply AxiomII in H9.
    destruct H9, H10; assert (A1 ⊂ (B∩A1) ∪ A).
    { unfold Included; intros; double H12; unfold Included in H7.
      apply H7 in H13; apply Theorem4 in H13; apply Theorem4.
      destruct H13; try tauto; left; apply Theorem4'; split; auto. }
    apply H2 with (A:= (B∩A1) ∪ A); auto.
Qed.

Lemma LemmaH1 : ∀ (f A: Class),
  FiniteSet f -> A∈f -> (∃ M, MaxMember M f /\ A ⊂ M).
Proof.
  intros; double H.
  assert (A ∈ (En_f1 f A)).
  { unfold En_f1; apply AxiomII; repeat split; Ens.
    rewrite Theorem5; auto. }
  assert ((En_f1 f A) ≠ ∅).
  { generalize (classic ((En_f1 f A) = ∅)); intros.
    destruct H3; auto; generalize (Theorem16 A); intros.
    rewrite H3 in H2; contradiction. }
  apply LemmaH1_1 with (A:=A) in H1; auto.
  add ((En_f1 f A) ≠ ∅) H1; apply Tu.Tukey in H1.
  destruct H1 as [M H1]; exists M; unfold MaxMember in H1.
  double H3; apply H1 in H4; clear H1; destruct H4.
  assert ((M ∪ A) ∈ (En_f1 f A)).
  { unfold En_f1; apply AxiomII.
    unfold En_f1 in H1; apply AxiomII in H1; destruct H1, H5.
    unfold En_f1 in H2; apply AxiomII in H2; destruct H2, H7.
    repeat split; try (apply AxiomIV; split); auto.
    rewrite Theorem7; rewrite Theorem5; auto. }
  apply H4 in H5; unfold ProperIncluded in H5.
  apply not_and_or in H5; destruct H5.
  - cut (M ⊂ M ∪ A); intros; try contradiction.
    unfold Included; intros; apply Theorem4; auto.
  - apply NNPP in H5; assert (A ⊂ M).
    { rewrite H5; unfold Included; intros; apply Theorem4; auto. }
    split; auto; unfold MaxMember; unfold FiniteSet in H; destruct H.
    unfold En_f1 in H1; apply AxiomII in H1; destruct H1, H8; intros.
    clear H10; repeat split; auto; intro K; intros; intro.
    unfold ProperIncluded in H11; destruct H11.
    add (M ⊂ K) H6; apply Theorem28 in H6; apply Theorem29 in H6.
    double H10; rewrite Theorem6 in H6; rewrite <- H6 in H13.
    assert (K ∈ (En_f1 f A)).
    { unfold En_f1; apply AxiomII; repeat split; Ens. }
```

```
    apply H4 in H14; unfold ProperIncluded in H14; tauto.
Qed.

Hint Resolve LemmaH1 : Axiom_of_Choice.

(* LemmaH2 *)

Lemma LemmaH2 : ∀ (A: Class),
  Ensemble A -> FiniteSet \{ λ n, n ⊂ A /\ Nest n \}.
Proof.
  intros.
  unfold FiniteSet; repeat split; intros.
  - apply Theorem38 in H; auto.
    assert (\{ λ n, n ⊂ A /\ Nest n \} ⊂ pow(A)).
    { unfold Included at 1; intros.
      unfold PowerSet; apply AxiomII.
      apply AxiomII in H0; destruct H0, H1; split; auto. }
    apply Theorem33 in H0; auto.
  - apply AxiomII in H0; apply AxiomII; destruct H0, H1, H2.
    double H1; add (F ⊂ A) H5; apply Theorem28 in H5; double H5.
    apply Theorem33 in H6; repeat split; auto; intros; unfold Nest.
    intros; unfold Nest in H4; destruct H7; unfold Included in H1.
    apply H1 in H7; apply H1 in H8; add (B∈F) H7.
  - destruct H0; apply AxiomII; repeat split; auto; intros.
    + unfold Included; intros; assert ([z] ⊂ F /\ Finite ([z])).
      { split; try (apply Finite_Single; Ens).
        unfold Included; intros; apply AxiomII in H3.
        destruct H3; rewrite H4; auto; apply Theorem19; Ens. }
      apply H1 in H3; apply AxiomII in H3; destruct H3, H4.
      unfold Included in H4; apply H4; apply AxiomII; Ens.
    + unfold Nest; intros; destruct H2.
      assert ([A0|B] ⊂ F /\ Finite ([A0|B])).
      { split.
        - unfold Included; intros; unfold Unordered in H4.
          apply Theorem4 in H4; destruct H4.
          + unfold Singleton in H4; apply AxiomII in H4.
            destruct H4; rewrite H5; auto; apply Theorem19; Ens.
          + unfold Singleton in H4; apply AxiomII in H4.
            destruct H4; rewrite H5; auto; apply Theorem19; Ens.
        - unfold Unordered; apply Theorem168.
          split; apply Finite_Single; Ens. }
      apply H1 in H4; apply AxiomII in H4; destruct H4, H5.
      unfold Nest in H6; apply H6; split.
      * unfold Unordered; apply Theorem4; left.
        unfold Singleton; apply AxiomII; Ens.
```

```
    * unfold Unordered; apply Theorem4; right.
      unfold Singleton; apply AxiomII; Ens.
Qed.

Hint Resolve LemmaH2 : Axiom_of_Choice.

(* Hausdorff Maximum Principle*)

Theorem Hausdorff : ∀ (A N: Class),
  Ensemble A -> N ⊂ A /\ Nest N ->
  (∃ u, MaxMember u \{ λ n, n⊂A /\ Nest n \} /\ N ⊂ u).
Proof.
  intros; destruct H0; double H.
  apply LemmaH2 in H2; double H0; apply Theorem33 in H0; auto.
  apply LemmaH1 with (A:= N) in H2; auto.
  apply AxiomII; auto.
Qed.

Hint Resolve Hausdorff : Axiom_of_Choice.

End Husdorff_Principle.
```

4.4 极大原则

定理 (极大原则)　*设 $\mathcal{A}$ 是一个集合, 如果对于 $\mathcal{A}$ 中的每一个套 $\mathcal{N}$, 存在 $\mathcal{A}$ 中某一个元素 N 包含着 $\mathcal{N}$ 中的任何一个元素, 则 $\mathcal{A}$ 中必有极大成员.*

极大原则通过 Hausdorff 极大原则容易证明, 其 Coq 描述及证明代码如下:

```
Require Export Hausdorff_Maximal_Principle.

(** Maximal Principle **)

Module Type Maximal_Principle.

Declare Module Hus : Husdorff_Principle.

Lemma Ex_Nest : ∀ A, ∃ N, N ⊂ A /\ Nest N.
Proof.
  intros.
  exists ∅; split; try apply Theorem26; unfold Nest; intros.
  destruct H; generalize (Theorem16 B); contradiction.
Qed.
```

```
Theorem MaxPrinciple : ∀ (A: Class), Ensemble A ->
  (∀ n, n ⊂ A /\ Nest n -> ∃ N, N ∈ A /\ (∀ u, u ∈ n -> u ⊂ N)) ->
  ∃ M, MaxMember M A.
Proof.
  intros.
  generalize (Ex_Nest A); intros; destruct H1 as [n H1].
  assert (\{ λ n, n ⊂ A /\ Nest n \} ≠ ∅).
  { apply Property_NotEmpty; exists n; destruct H1.
    apply AxiomII; repeat split; auto; apply Theorem33 in H1; auto. }
  apply Hus.Hausdorff in H1; auto; destruct H1 as [u H1], H1.
  unfold MaxMember in H1; apply H1 in H2; clear H1; destruct H2.
  apply AxiomII in H1; destruct H1; elim H4; intros; apply H0 in H4.
  destruct H4 as [N H4];exists N;unfold MaxMember;destruct H4;intro.
  clear H8; repeat split; auto; intro K; intros; intro.
  unfold ProperIncluded in H9; destruct H9.
  assert ((u ∪ [K]) ∈ \{ λ n, n⊂A /\ Nest n \}).
  { apply AxiomII; assert (Ensemble (u ∪ [K])).
    { apply AxiomIV; split; auto; apply Theorem42; Ens. }
    repeat split; auto; intros.
    - unfold Included; intros; apply Theorem4 in H12; destruct H12.
      + unfold Included in H5; apply H5 in H12; auto.
      + unfold Singleton in H12; apply AxiomII in H12.
        destruct H12; rewrite H13; auto; apply Theorem19; Ens.
    - unfold Nest; intros; destruct H12.
      apply Theorem4 in H12; apply Theorem4 in H13.
      assert (K ∈ 𝒰). { apply Theorem19; Ens. }
      unfold Nest in H6; destruct H12, H13.
      + add (B ∈ u) H12.
      + unfold Singleton in H13; apply AxiomII in H13.
        destruct H13; rewrite <- H15 in H9; auto; apply H7 in H12.
        add (N ⊂ B) H12; apply Theorem28 in H12; tauto.
      + unfold Singleton in H12; apply AxiomII in H12.
        destruct H12; rewrite <- H15 in H9; auto; apply H7 in H13.
        add (N ⊂ A0) H13; apply Theorem28 in H13; tauto.
      + unfold Singleton in H12, H13; apply AxiomII in H12.
        apply AxiomII in H13; destruct H12, H13; left.
        rewrite H15, H16; auto; unfold Included; auto. }
  apply H2 in H11; unfold ProperIncluded in H11.
  apply not_and_or in H11; destruct H11.
  - absurd (u ⊂ u ∪ [K]); auto.
    unfold Included; intros; apply Theorem4; auto.
  - apply NNPP in H11; assert (K∈u).
    { rewrite H11; apply Theorem4; right.
      unfold Singleton; apply AxiomII; split; Ens. }
```

```
    apply H7 in H12; add (K ⊂ N) H9.
    apply Theorem27 in H9; contradiction.
Qed.

Hint Resolve MaxPrinciple : Axiom_of_Choice.

End Maximal_Principle.
```

4.5 Zermelo 假定

定理 (Zermelo 假定) 设 $\mathcal{A}$ 是一个由非空集合构成的无交集合, 则存在一个集合 C 使得对于每一个 $A \in \mathcal{A}$, $A \cap C$ 是一个单点集.

Zermelo 假定由极大原则可以证明, 其 Coq 描述及证明代码如下:

```
Require Export Maximal_Principle.

(** Zermelo Postulate **)

Module Type Zermelo_Postulate.

Declare Module Max : Maximal_Principle.

Definition En_TB B A : Class :=
  \{ λ k, k ⊂ ∪A /\ (∀ a, a ∈ (A ∼ B) -> a ∩ k = ∅) /\
  (∀ a, a ∈ B -> ∃ x, a ∩ k = [x]) \}.

Definition En_T A : Class :=
  \{ λ k, ∃ B, B ⊂ A /\ k ∈ (En_TB B A) \}.

Theorem Zermelo : ∀ A,
  Ensemble A -> ∅ ∉ A ->
  (∀ x y, x ∈ A /\ y ∈ A -> x ≠ y -> x ∩ y = ∅) ->
  (∃ D, Ensemble D /\ (∀ B, B ∈ A -> ∃ x, B ∩ D = [x])).
Proof.
  intros.
  generalize (classic (A = ∅)); intros; destruct H2.
  - rewrite H2 in *; clear H2; exists ∅; split; intros; auto.
    generalize (Theorem16 B); intros; contradiction.
  - assert (Ensemble (En_T A)).
    { apply AxiomVI in H; apply Theorem38 in H; auto.
      assert (En_T A ⊂ pow(⋃A)).
      { unfold Included; intros; apply AxiomII in H3; destruct H3.
        destruct H4 as [B H4], H4; unfold En_TB in H5.
```

```
    apply AxiomII in H5; destruct H5, H6.
    unfold PowerSet; apply AxiomII; split; auto. }
  apply Theorem33 in H3; auto. }
assert (En_T A ≠ ∅).
{ apply Property_NotEmpty in H2; destruct H2.
  generalize (classic (x = ∅)); intros; destruct H4.
  - rewrite H4 in H2; contradiction.
  - apply Property_NotEmpty in H4; destruct H4.
    apply Property_NotEmpty; exists [x0]; unfold En_T.
    AssE x0; apply Theorem42 in H5; auto.
    apply AxiomII; split; auto; exists [x]; repeat split.
    + unfold Included, Singleton; intros; apply AxiomII in H6.
      destruct H6; rewrite H7; try apply Theorem19; Ens.
    + unfold En_TB; apply AxiomII; repeat split; auto; intros.
      * unfold Included; intros; apply AxiomII in H6; destruct H6.
        rewrite H7; try apply Theorem19; Ens; apply AxiomII; Ens.
      * unfold Setminus in H6; apply Theorem4' in H6; destruct H6.
        assert (x ∈ A /\ a ∈ A); auto; apply H1 in H8.
        { generalize (classic (a ∩ [x0] = ∅)); intros;
          destruct H9; auto.
          apply Property_NotEmpty in H9; destruct H9.
          apply Theorem4' in H9; destruct H9; apply AxiomII in H10.
          destruct H10; rewrite H11 in *; try apply Theorem19; Ens.
          assert (x0 ∈ (x ∩ a)); try apply Theorem4'; auto.
          rewrite H8 in H12; generalize (Theorem16 x0);
          contradiction. }
        { intro; rewrite H9 in H7; unfold Complement in H7.
          apply AxiomII in H7; destruct H7, H10; unfold Singleton.
          apply AxiomII; split; auto. }
      * exists x0; apply AxiomII in H6; destruct H6.
        rewrite H7 in *; try apply Theorem19; Ens.
        apply AxiomI; split; intros.
        -- apply Theorem4' in H8; apply H8.
        -- apply Theorem4'; split; auto; apply AxiomII in H8.
           destruct H8; rewrite H9; try apply Theorem19; Ens. }
apply Max.MaxPrinciple in H3.
+ destruct H3 as [C H3]; unfold MaxMember in H3.
  apply H3 in H4; clear H3; destruct H4; exists C; split; Ens.
  unfold En_T in H3; apply AxiomII in H3; destruct H3.
  destruct H5 as [B H5], H5.
  generalize (classic (B = A)); intro; destruct H7.
  * rewrite H7 in H6; apply AxiomII in H6; apply H6.
  * double H5; apply Property_∅ in H8; assert (A ~ B <> ∅).
    { intro; apply H8 in H9; symmetry in H9; tauto. }
    clear H7 H8; apply Property_NotEmpty in H9; destruct H9.
```

```
unfold Setminus in H7; apply Theorem4' in H7; destruct H7.
generalize (classic (x = ∅)); intro; destruct H9;
try (rewrite H9 in H7; contradiction).
apply Property_NotEmpty in H9; destruct H9.
assert ((C ∪ [x0]) ∈ (En_T A)).
{ unfold En_T; apply AxiomII; split.
  - apply AxiomIV; split; try apply Theorem42; Ens.
  - exists (B ∪ [x]); repeat split.
    + unfold Included; intros; apply Theorem4 in H10.
      destruct H10; auto; apply AxiomII in H10; destruct H10.
      rewrite H11; try apply Theorem19; Ens.
    + apply AxiomII; repeat split; intros.
      * apply AxiomIV; split; try apply Theorem42; Ens.
      * unfold Included; intros; apply Theorem4 in H10;
        destruct H10.
        { apply AxiomII in H6; apply H6 in H10; auto. }
        { unfold Singleton in H10; apply AxiomII in H10;
          destruct H10.
          rewrite H11; try apply Theorem19; Ens;
          unfold Element_U.
          apply AxiomII; split; Ens. }
      * apply AxiomII in H6; unfold Setminus in H10.
        apply Theorem4' in H10; destruct H10; rewrite Theorem8.
        unfold Complement in H11; apply AxiomII in H11;
        destruct H11.
        assert (a ∈ (B ∪ [x]) <-> a ∈ B \/ a ∈ [x]).
        { split; try apply Theorem4. }
        apply Lemma_z in H13; auto; clear H12.
        apply not_or_and in H13; destruct H13.
        assert (a ∈ (A ~ B)).
        { unfold Setminus; apply Theorem4'; split; auto.
          unfold Complement; apply AxiomII; auto. }
        apply H6 in H14; rewrite H14; clear H14.
        assert (x ∈ A /\ a ∈ A); auto; apply H1 in H14.
        { generalize (classic (a ∩ [x0] = ∅)); intros;
          destruct H15.
          - rewrite H15; apply Theorem5.
          - apply Property_NotEmpty in H15; destruct H15.
            apply Theorem4' in H15; destruct H15;
            apply AxiomII in H16.
            destruct H16; rewrite H17 in *; try apply Theorem19;
            Ens.
            assert (x0 ∈ (x ∩ a)); try apply Theorem4'; auto.
            rewrite H14 in H18; destruct (Theorem16 _ H18). }
        { intro; rewrite H15 in H13; destruct H13;
```

```
            unfold Singleton.
            apply AxiomII; split; auto. }
        * apply AxiomII in H6; rewrite Theorem8.
          apply Theorem4 in H10; destruct H10.
          { double H10; apply H6 in H11; destruct H11; rewrite H11.
            clear H11; assert (x ∈ A /\ a ∈ A); auto;
            apply H1 in H11.
            - generalize (classic (a ∩ [x0] = ∅)); intros;
              destruct H12.
              + rewrite H12; rewrite Theorem6, Theorem17; Ens.
              + apply Property_NotEmpty in H12; destruct H12.
                apply Theorem4' in H12; destruct H12;
                apply AxiomII in H13.
                destruct H13; rewrite H14 in *;
                try apply Theorem19; Ens.
                assert (x0 ∈ (x ∩ a)); try apply Theorem4'; auto.
                rewrite H11 in H15; destruct (Theorem16 _ H15).
            - intro; rewrite <- H12 in H10;
              unfold Complement in H8.
              apply AxiomII in H8; destruct H8; contradiction. }
          { unfold Singleton in H10; apply AxiomII in H10;
            destruct H10.
            rewrite H11; try apply Theorem19; Ens; clear H10 H11 a.
            assert (x ∈ (A ~ B)); try apply Theorem4'; auto.
            apply H6 in H10; rewrite H10, Theorem17; exists x0.
            apply AxiomI; split; intros.
            - apply Theorem4' in H11; apply H11.
            - apply Theorem4'; split; auto; apply AxiomII in H11.
              destruct H11; rewrite H12;
              try apply Theorem19; Ens. } }
    apply H4 in H10; destruct H10; unfold ProperIncluded.
    split; unfold Included; intros; try (apply Theorem4; auto).
    apply AxiomII in H6; assert (x ∈ (A ~ B));
    try apply Theorem4'; auto.
    apply H6 in H10; intro; assert (x0 ∈ ∅).
    { rewrite <- H10; apply Theorem4'; split; auto; rewrite H11.
      apply Theorem4; right; apply AxiomII; split; Ens. }
    generalize (Theorem16 x0); intros; contradiction.
  + intros; exists (⋃n); split; intros.
    * apply Property_FinSet in H5; auto; split; auto; clear H6.
      destruct H5; unfold FiniteSet; repeat split; auto.
      { intros F H7 F1 H8; destruct H8; unfold En_T in H7.
        apply AxiomII in H7; destruct H7, H10 as [B H10], H10.
        unfold En_T; apply AxiomII; assert (Ensemble F1).
        { apply Theorem33 in H8; auto. }
```

```
split; auto; unfold En_TB in H11; apply AxiomII in H11.
destruct H11 as [_ H11], H11.
exists \{ λ a, a ∈ B /\ ∃ x, a ∩ F1 = [x] \}; repeat split.
- unfold Included; intros; apply AxiomII in H14;
  destruct H14, H15.
  apply H10 in H15; auto.
- apply AxiomII; repeat split; auto; intros.
  + apply Theorem28 with (y:= F); split; auto.
  + apply Theorem4'in H14; destruct H14; apply AxiomII in H15.
    destruct H15 as [_ H15]; unfold NotIn in H15.
    apply Lemma_z with (B:= Ensemble a /\
    a∈B /\ (∃x, a∩F1=[x])) in H15.
    * apply not_and_or in H15; destruct H15.
      { destruct H15; Ens. }
      { apply not_and_or in H15; destruct H15.
        - assert (a ∈ (A ~ B)).
          { unfold Setminus; apply Theorem4'; split; auto.
            unfold Complement; apply AxiomII; split; Ens. }
          apply H13 in H16; apply Theorem27; split;
          try apply Theorem26.
          rewrite <- H16; unfold Included; intros.
          apply Theorem4' in H17; apply Theorem4';
          destruct H17; Ens.
        - generalize (classic (a ∈ B)); intros; destruct H16.
          + apply H13 in H16; destruct H16.
            apply not_ex_all_not with (n:= x) in H15.
            generalize (classic (a∩F1=∅)); intros;
            destruct H17; auto.
            apply Property_NotEmpty in H17;
            destruct H17 as [z H17].
            assert ((a ∩ F1) ⊂ [x]).
            { unfold Included; intros; rewrite <- H16.
              apply Theorem4' in H18; apply Theorem4';
              destruct H18.
              split; auto. }
            double H17; apply H18 in H19;
            unfold Singleton in H19.
            apply AxiomII in H19; destruct H19.
            assert ([x] ⊂ (a ∩ F1)).
            { unfold Included; intros; unfold Singleton in H21.
              apply AxiomII in H21; destruct H21.
              assert (Ensemble x).
              { apply Theorem42'; rewrite <- H16.
                apply Theorem33 with (x:= a); Ens;
                unfold Included.
```

```
              intros; apply Theorem4' in H23; apply H23. }
            rewrite H22, <- H20; try apply Theorem19; Ens. }
          assert ((a ∩ F1) ⊂ [x] /\ [x] ⊂ (a ∩ F1)); auto.
          apply Theorem27 in H22; rewrite H22 in H15; tauto.
        + assert (a ∈ (A ~ B)).
          { unfold Setminus; apply Theorem4'; split; auto.
            unfold Complement; apply AxiomII; split; Ens. }
          apply H13 in H17; apply Theorem27.
          split; try apply Theorem26; rewrite <- H17.
          unfold Included; intros; apply Theorem4' in H18.
          apply Theorem4'; destruct H18; Ens. }
    * split; intros; try apply AxiomII in H16;
      try apply AxiomII; Ens.
  + apply AxiomII in H14; apply H14. }
{ intros; destruct H7.
  generalize (classic (F ∈ (En_T A))); intros;
  destruct H9; auto.
  assert (forall B, B ⊂ A -> ~ F ∈ (En_TB B A)).
  { intros; unfold En_T in H9.
    apply Lemma_z with (B:= Ensemble F /\ ∃ B, B ⊂ A /\
    F ∈ (En_TB B A)) in H9.
    - apply not_and_or in H9; destruct H9; try tauto.
      apply not_ex_all_not with (n:= B) in H9.
      apply not_and_or in H9; tauto.
    - split; intros; try apply AxiomII; auto.
      apply AxiomII in H11; apply H11. }
  set (B:= \{ λ a, a ∈ A /\ a ∩ F <> ∅ /\
  ~ (∃x, a ∩ F = [x]) \} ∪
  \{ λ a, a ∈ A /\ a ∩ F <> ∅ /\ (∃x, a ∩ F = [x]) \}).
  assert (B ⊂ A).
  { unfold Included; intros; apply Theorem4 in H11.
    destruct H11; apply AxiomII in H11; apply H11. }
  apply H10 in H11; unfold En_TB in H11.
  apply Lemma_z with (B:= Ensemble F /\ F ⊂ ⋃ A /\
  (∀a, a ∈ (A ~ B) ->
  a ∩ F = ∅) /\ (∀a, a ∈ B -> ∃x, a ∩ F = [x])) in H11.
  - apply not_and_or in H11; destruct H11; try tauto.
    apply not_and_or in H11; destruct H11.
    + assert (∀ z, z ⊂ F /\ Finite z -> z ⊂ ⋃ A).
      { intros; apply H8 in H12; unfold En_T in H12.
        apply AxiomII in H12; destruct H12, H13, H13.
        unfold En_TB in H14; apply AxiomII in H14; apply H14. }
      destruct H11; unfold Included; intros.
      assert ([z] ⊂ F /\ Finite ([z])).
      { split; try apply Finite_Single; Ens;
```

```
    unfold Included; intros.
    unfold Singleton in H13; apply AxiomII in H13;
    destruct H13.
    rewrite H14; try apply Theorem19; Ens. }
  apply H12 in H13; apply H13; unfold Singleton.
  apply AxiomII; split; Ens.
+ apply not_and_or in H11; destruct H11.
  * apply not_all_ex_not in H11; destruct H11 as [a H11].
    apply imply_to_and in H11; destruct H11.
    unfold Setminus in H11; apply Theorem4' in H11;
    destruct H11.
    unfold Complement in H13; apply AxiomII in H13;
    destruct H13.
    unfold NotIn in H14;
    apply Lemma_z with (B:= a ∈ \{λ a, a∈A /\
    a ∩ F <> ∅ /\ ~ (∃ x, a ∩ F = [x])\} \/
    a ∈ \{ λ a, a ∈ A /\
    a ∩ F <> ∅ /\ (∃ x, a ∩ F = [x]) \}) in H14.
    { apply not_or_and in H14; destruct H14.
      apply Lemma_z with (B:= Ensemble a /\ a ∈ A /\ a ∩ F
      <> ∅ /\ ~ (∃ x, a ∩ F = [x])) in H14.
      - apply Lemma_z with (B:= Ensemble a /\ a ∈ A /\ a ∩
        F <> ∅ /\ (∃ x, a ∩ F = [x])) in H15; try tauto.
        split; intros; try apply AxiomII; auto.
        apply AxiomII in H16; auto.
      - split; intros; try apply AxiomII; auto.
        apply AxiomII in H16; auto. }
    { split; intros; try apply Theorem4; auto. }
  * apply not_all_ex_not in H11; destruct H11 as [a H11].
    apply imply_to_and in H11; destruct H11.
    apply Theorem4 in H11; destruct H11.
    { apply AxiomII in H11; clear B H12;
      destruct H11, H12, H13.
      assert (~ (forall x y, x ∈ (a ∩ F) /\ y ∈ (a ∩ F) ->
      x = y)).
      { intro; apply Property_NotEmpty in H13;
        destruct H13, H14.
        exists x; apply AxiomI; split; intros.
        - add (x ∈ (a ∩ F)) H14; apply H15 in H14;
          rewrite H14.
          unfold Singleton; apply AxiomII; Ens.
        - unfold Singleton in H14;
          apply AxiomII in H14.
          destruct H14; rewrite H16;
          try apply Theorem19; Ens. }
```

```
destruct H15; intros; destruct H15.
generalize (classic (x = y)); intros;
destruct H17; auto.
apply Theorem4' in H15; apply Theorem4' in H16.
destruct H15, H16.
assert ([x|y] ⊂ F /\ Finite ([x|y])).
{ split.
  - unfold Included; intros;
    apply Theorem46 in H20; Ens.
    destruct H20; rewrite H20; auto.
  - unfold Unordered; apply Theorem168.
    split; apply Finite_Single; Ens. }
apply H8 in H20; unfold En_T in H20;
apply AxiomII in H20.
destruct H20,H21 as [B H21],H21;unfold En_TB in H22.
apply AxiomII in H22; destruct H22, H23.
generalize (classic (a ∈ B)); intros; destruct H25.
- apply H24 in H25; destruct H25.
  assert ([x | y] ⊂ a).
  { unfold Included; intros;
    apply Theorem46 in H26; Ens.
    destruct H26; rewrite H26; auto. }
  apply Theorem30 in H26; rewrite Theorem6' in H26.
  rewrite H26 in H25; clear H26; destruct H17.
  assert (x ∈ [x0] /\ y ∈ [x0]).
  { rewrite <- H25; split; apply Theorem46; Ens. }
  destruct H17; apply AxiomII in H17.
  apply AxiomII in H26; destruct H17, H26.
  assert (x0 ∈ ∩U).
  { apply Theorem19; apply Theorem42'; rewrite <- H25.
    apply Theorem46; Ens. }
  rewrite H27, H28; auto.
- assert (a ∈ (A ~ B)).
  { unfold Setminus; apply Theorem4'; split; auto.
    unfold Complement; apply AxiomII; split; Ens. }
  apply H24 in H26. clear H24.
  assert ([x | y] ⊂ a).
  { unfold Included; intros;
    apply Theorem46 in H24; Ens.
    destruct H24; rewrite H24; auto. }
  apply Theorem30 in H24; rewrite Theorem6' in H24.
  rewrite H24 in H26; clear H24.
  assert (x ∈ ∅).
  { rewrite <- H26; apply Theorem46; Ens. }
  generalize (Theorem16 x); intros; contradiction. }
```

```
            { apply AxiomII in H11; destruct H11, H13, H14; tauto. }
        - split; intros; try apply AxiomII;
          try apply AxiomII in H12; auto. }
      * unfold Included; intros; apply AxiomII; split; Ens.
Qed.

Hint Resolve Zermelo : Axiom_of_Choice.

End Zermelo_Postulate.
```

4.6 Zorn 引理

定理 (Zorn 引理) 设 $(X, \preceq)$ 是一个偏序集. 如果 X 中每一个链都有上界, 则 X 中必有极大元素.

Zorn 引理由极大原则可以证明, 其 Coq 描述及证明代码如下:

```
Require Export Maximal_Principle.

Module Type Zorn_Lemma.

Declare Module Max : Maximal_Principle.

Definition En_Fs X le x := \{ λ u, u ∈ X /\ Rrelation u le x \}.

Definition En_FF X A le := \{ λ u, ∃ a, u = (En_Fs X le a)
  /\ a ∈ A \}.
Axiom Property_FF : ∀ X A le a,
  (En_Fs X le a) ∈ (En_FF X A le) ->
  (∀ b, En_Fs X le a = En_Fs X le b -> a = b).

Definition En_A X A le := \{ λ u, (En_Fs X le u) ∈ A \}.

Lemma Property_Fs : ∀ X le u v,
  PartialOrder le X -> u ∈ X ->
  (En_Fs X le u ⊂ En_Fs X le v <-> Rrelation u le v).
Proof.
  intros; unfold PartialOrder in H.
  destruct H, H1, H2, H3; split; intros.
  - assert (u ∈ (En_Fs X le u)).
    { unfold En_Fs; apply AxiomII; repeat split; Ens. }
    unfold Included in H5; apply H5 in H6; unfold En_Fs in H6.
    apply AxiomII in H6; try apply H6.
```

```
  - unfold Included; intros; unfold En_Fs in H6; unfold En_Fs.
    apply AxiomII in H6; apply AxiomII; destruct H6, H7.
    repeat split; auto; unfold Transitivity in H4; apply H4 with (v:=u).
    repeat split; auto; destruct H1; unfold Included in H9.
    unfold Rrelation in H5; apply H9 in H5; unfold Cartesian in H5.
    apply AxiomII_P in H5; apply H5.
Qed.

Lemma LemmaZorn_1 : ∀ A X le,
  PartialOrderSet X le -> (Chain A X le <->
  (En_FF X A le) ⊂ (En_FF X X le) /\ Nest(En_FF X A le) /\ A ≠ ∅).
Proof.
  intros; split; intros.
  - double H; unfold PartialOrderSet in H1; unfold Chain in H0.
    apply H0 in H1; clear H0; destruct H1, H0; split.
    + unfold Included; intros; apply AxiomII in H3; apply AxiomII.
      destruct H3, H4, H4; split; auto; exists x; split; auto.
    + split; auto; unfold Nest; intros; destruct H3;
      apply AxiomII in H3.
      apply AxiomII in H4; destruct H3, H4, H5, H5, H6, H6.
      generalize (classic (x = x0)); intros; destruct H9.
      * rewrite H9 in H5; rewrite H5, H6.
        right; unfold Included; intros; auto.
      * unfold TotalOrder in H1; destruct H1; clear H1; double H7.
        add (x0 ∈ A) H7; apply H10 in H7; auto; clear H10; clear H9.
        unfold PartialOrderSet in H; unfold PartialOrder in H.
        destruct H, H9, H10 as [H11 H10], H10 as [H12 H10];
        clear H11 H12.
        unfold Transitivity in H10; rewrite H5, H6; destruct H7.
        -- left; unfold Included; intros; apply AxiomII in H11.
           destruct H11, H12; apply AxiomII; repeat split; auto.
           assert (Rrelation x le x0).
           { unfold Rrelation in H7; unfold Rrelation.
             apply Theorem4' in H7; apply H7. }
           unfold Included in H0; apply H0 in H1; apply H0 in H8.
           apply H10 with (v:= x); auto.
        -- right; unfold Included; intros; apply AxiomII in H11.
           destruct H11, H12; apply AxiomII; repeat split; auto.
           assert (Rrelation x0 le x).
           { unfold Rrelation in H7; unfold Rrelation.
             apply Theorem4' in H7; apply H7. }
           unfold Included in H0; apply H0 in H1; apply H0 in H8.
           apply H10 with (v:= x0); auto.
  - destruct H0, H1; unfold Chain; intros; clear H3.
    assert (∀ z, z∈A -> (En_Fs X le z) ∈ (En_FF X A le)).
```

```
{ intros; unfold En_FF; apply AxiomII; split.
  - assert (En_Fs X le z ⊂ X).
    { unfold Included; intros; apply AxiomII in H4; apply H4. }
    unfold PartialOrderSet in H; destruct H; clear H5.
    apply Theorem33 in H4; auto.
  - exists z; split; auto. }
assert (A ⊂ X); try split; auto.
{ unfold Included; intros; unfold Included in H0.
  apply H3 in H4; apply H0 in H4; double H4; unfold En_FF in H5.
  apply AxiomII in H5; destruct H5, H6, H6.
  apply Property_FF with (A:= X) in H6; auto; rewrite H6; auto. }
unfold PartialOrderSet in H; unfold PartialOrder in H; destruct H.
destruct H5, H6, H7; unfold TotalOrder.
split; try (apply Theorem33 with (x:= X); auto).
unfold PartialOrder; repeat split; auto.
+ apply Theorem33 in H4; auto.
+ destruct H5; unfold Relation in H5; unfold Relation; intros.
  apply Theorem4' in H10; destruct H10; apply H5 in H10; auto.
+ unfold Included; intros; apply Theorem4' in H9; apply H9.
+ unfold Reflexivity; intros; unfold Rrelation; apply Theorem4'.
  unfold Reflexivity in H6; double H9; unfold Included in H4.
  apply H4 in H10; apply H6 in H10; unfold Rrelation in H10.
  split; auto; unfold Cartesian; apply AxiomII_P; Ens.
+ unfold Antisymmetry in H7; unfold Antisymmetry; intros.
  destruct H9, H10; apply H7; auto; unfold Rrelation in H10, H12.
  apply Theorem4' in H10; apply Theorem4' in H12; destruct H10, H12.
  unfold Rrelation; split; auto.
+ unfold Transitivity in H8; unfold Transitivity; intros.
  destruct H9, H10, H9, H12; assert ((u∈X /\ v∈X /\ w∈X) /\
  Rrelation u le v /\ Rrelation v le w).
  { unfold Included in H4; apply H4 in H9; apply H4 in H12.
    apply H4 in H13; repeat split; auto; unfold Rrelation.
    - unfold Rrelation in H10; apply Theorem4' in H10; apply H10.
    - unfold Rrelation in H11; apply Theorem4' in H11; apply H11. }
  apply H8 in H14; unfold Rrelation in H14; clear H8.
  unfold Rrelation; apply Theorem4'; split; auto.
  unfold Cartesian; apply AxiomII_P; Ens.
+ intros; destruct H9; unfold Nest in H1.
  assert ((En_Fs X le x) ∈ (En_FF X A le) /\ (En_Fs X le y) ∈
  (En_FF X A le)). { apply H3 in H9; apply H3 in H11; split; auto. }
  apply H1 in H12; clear H1; clear H2; destruct H12.
  * { left; unfold Rrelation; apply Theorem4'; split.
      - apply Property_ProperIncluded in H1; destruct H1.
        + unfold ProperIncluded in H1; destruct H1; clear H2.
          apply Property_Fs in H1; unfold Rrelation in H1; auto.
```

```
            unfold PartialOrder; auto.
          + assert ((En_Fs X le x) ∈ (En_FF X A le)).
            { unfold En_FF; apply AxiomII; double H9.
              apply H3 in H11; split; Ens. }
            apply Property_FF with (A:= A) in H1; auto; contradiction.
        - unfold Cartesian; apply AxiomII_P; repeat split; auto.
          apply Theorem49; split; Ens. }
    * { right; unfold Rrelation; apply Theorem4'; split.
        - apply Property_ProperIncluded in H1; destruct H1.
          + unfold ProperIncluded in H1; destruct H1; clear H2.
            apply Property_Fs in H1; unfold Rrelation in H1; auto.
            unfold PartialOrder; auto.
          + assert ((En_Fs X le y) ∈ (En_FF X A le)).
            { unfold En_FF; apply AxiomII; double H9.
              apply H3 in H11; split; Ens. }
            apply Property_FF with (A:= A) in H1; auto.
            symmetry in H1; contradiction.
        - unfold Cartesian; apply AxiomII_P; repeat split; auto.
          apply Theorem49; split; Ens. }
Qed.

Lemma LemmaZorn_2 : ∀ A X le y,
  PartialOrderSet X le /\ X ≠ ∅ ->
  (BoundU y A X le -> (∀x, x∈(En_FF X A le) -> x⊂(En_Fs X le y))).
Proof.
  intros; double H.
  unfold BoundU in H0; apply H0 in H2; clear H0; destruct H2, H2.
  unfold En_FF in H1; apply AxiomII in H1; destruct H1.
  destruct H4 as [a H4], H4; rewrite H4; unfold Included; intros.
  unfold En_Fs in H6; unfold En_Fs; apply AxiomII in H6; apply AxiomII.
  double H5; apply H3 in H5; destruct H6, H8, H; clear H10.
  repeat split; auto; unfold PartialOrderSet, PartialOrder in H.
  destruct H, H10, H11, H12; unfold Transitivity in H13.
  apply H13 with (v:=a); repeat split; auto.
Qed.

Lemma LemmaZorn_3 : ∀ X le y,
  PartialOrderSet X le /\ X ≠ ∅ ->
  (MaxElement y X le <-> MaxMember (En_Fs X le y) (En_FF X X le)).
Proof.
  intros; destruct H; split; intros.
  - unfold MaxElement in H1; apply H1 in H0; clear H1; destruct H0.
    unfold PartialOrderSet, PartialOrder in H; destruct H.
    unfold MaxMember; intros; clear H3; repeat split; intros.
    + unfold En_FF; apply AxiomII; split.
```

```
      * assert ((En_Fs X le y) ⊂ X).
        { unfold Included; intros; apply AxiomII in H3; apply H3. }
        apply Theorem33 in H3; auto.
      * exists y; split; auto.
    + apply AxiomII in H3; destruct H3, H4, H4; rewrite H4; intro.
      apply H1 in H5; apply not_and_or in H5; destruct H5.
      * unfold ProperIncluded in H6; destruct H6; clear H7.
        apply Property_Fs in H6; unfold Rrelation in H6;
        try apply H2; auto.
        unfold PartialOrder; auto.
      * apply NNPP in H5;rewrite H5 in H6;unfold ProperIncluded in H6.
        destruct H6; contradiction.
  - unfold PartialOrderSet, PartialOrder in H; destruct H.
    unfold MaxElement; intros; clear H3.
    unfold MaxMember in H1; assert (En_FF X X le ≠ ∅).
    { apply Property_NotEmpty in H0; destruct H0.
      apply Property_NotEmpty; exists (En_Fs X le x).
      unfold En_FF; apply AxiomII; split.
      - apply Theorem33 with (x:= X); auto; unfold Included; intros.
        unfold En_Fs in H3; apply AxiomII in H3; apply H3.
      - exists x; split; auto. }
    apply H1 in H3; clear H1; clear H0; destruct H3; double H0.
    unfold En_FF in H3; apply AxiomII in H3; destruct H3, H4, H4.
    apply Property_FF with (A:= X) in H4; auto; rewrite <- H4 in H5.
    split; intros; auto; assert (En_Fs X le y0 ∈ (En_FF X X le)).
    { unfold En_FF; apply AxiomII; split.
      - assert (En_Fs X le y0 ⊂ X).
        { unfold Included; intros; apply AxiomII in H7; apply H7. }
        apply Theorem33 in H7; auto.
      - exists y0; split; auto. }
    apply H1 in H7; intro; elim H7; destruct H8.
    unfold ProperIncluded; split.
    + apply Property_Fs; auto; unfold PartialOrder; auto.
    + intro; apply Property_FF with (A:=X) in H10; auto.
Qed.

Theorem Zorn : ∀ X le,
  PartialOrderSet X le -> (∀ A, Chain A X le -> ∃ y, BoundU y A X le)
  -> ∃ v, MaxElement v X le.
Proof.
  intros.
  generalize (classic (X = ∅)); intros; destruct H1.
  - exists ∅; rewrite H1; unfold MaxElement; contradiction.
  - assert (Ensemble (En_FF X X le)).
    { unfold PartialOrderSet, PartialOrder in H; destruct H.
```

```
    clear H2; assert ((En_FF X X le) ⊂ pow(X)).
    { unfold Included; intros; unfold En_FF in H2.
      apply AxiomII in H2; destruct H2, H3, H3.
      unfold PowerSet; apply AxiomII; split; auto.
      rewrite H3; unfold Included; intros.
      unfold En_Fs in H5; apply AxiomII in H5; apply H5. }
    apply Theorem38 in H; auto; apply Theorem33 in H2; auto. }
  apply Max.MaxPrinciple in H2.
  + destruct H2 as [M H2]; double H2; unfold MaxMember in H3.
    assert (En_FF X X le ≠ ∅).
    { apply Property_NotEmpty in H1; destruct H1.
      apply Property_NotEmpty; exists (En_Fs X le x).
      unfold En_FF; apply AxiomII; split.
      - destruct H; apply Theorem33 with (x:= X); auto.
        unfold Included; intros; apply AxiomII in H5; apply H5.
      - exists x; split; auto. }
    apply H3 in H4; clear H3; destruct H4.
    unfold En_FF in H3; apply AxiomII in H3; destruct H3.
    destruct H5 as [v H5], H5; rewrite H5 in H2; add (X ≠ ∅) H.
    exists v; apply LemmaZorn_3 with (y:=v) in H; auto; apply H; auto.
  + intro B; intros; destruct H3.
    generalize (classic (B = ∅)); intros; destruct H5.
    { rewrite H5; apply Property_NotEmpty in H1.
      destruct H1 as [z H1]; exists (En_Fs X le z); split.
      - unfold En_FF; apply AxiomII; split.
        + destruct H; apply Theorem33 with (x:=X); auto.
          unfold Included; intros; apply AxiomII in H7; apply H7.
        + exists z; split; auto.
      - intros; generalize (Theorem16 u); contradiction. }
    assert ( (En_FF X (En_A X B le) le) ⊂ (En_FF X X le) /\
    Nest (En_FF X (En_A X B le) le) /\ (En_A X B le) ≠ ∅).
    { assert ((En_FF X (En_A X B le) le) ⊂ (En_FF X X le)).
      { intros; unfold Included; intros.
        apply AxiomII in H6; destruct H6, H7 as [a H7], H7.
        apply AxiomII in H8; destruct H8; apply AxiomII; split; auto.
        unfold Included in H3; apply H3 in H9; unfold En_FF in H8.
        double H9; apply AxiomII in H10; destruct H10, H11 as [a1 H11].
        destruct H11; apply Property_FF with (b:= a1) in H9; auto.
        rewrite H9 in H7; exists a1; auto. }
      repeat split; auto; unfold Nest; intros.
      - destruct H7; apply AxiomII in H7; apply AxiomII in H8.
        destruct H7, H8, H9 as [a H9], H10 as [b H10], H9, H10.
        apply AxiomII in H11; apply AxiomII in H12; destruct H11, H12.
        rewrite H9, H10; unfold Nest in H4;
        add ((En_Fs X le b) ∈ B) H13.
```

```
        - apply Property_NotEmpty in H5; destruct H5; double H5.
          apply H3 in H7; apply AxiomII in H7; destruct H7, H8, H8.
          rewrite H8 in H5; apply Property_NotEmpty; exists x0.
          apply AxiomII; split; Ens. }
      double H; apply LemmaZorn_1 with (A:= (En_A X B le)) in H7.
      apply H7 in H6; clear H7; double H6; apply H0 in H6; clear H0.
      destruct H6 as [y H0]; unfold Chain in H7; double H.
      unfold PartialOrderSet in H6; apply H7 in H6; clear H7.
      destruct H6; exists (En_Fs X le y); split; intros.
      * unfold BoundU in H0; double H; add (X≠∅) H8; apply H0 in H8.
        clear H0; destruct H8, H8 as [H9 H8]; clear H9.
        unfold En_FF; apply AxiomII; split.
        -- assert (En_Fs X le y ⊂ X).
           { unfold Included; intros; apply AxiomII in H9; apply H9. }
           destruct H; apply Theorem33 in H9; auto.
        -- exists y; split; auto.
      * double H8; unfold Included in H3; apply H3 in H9.
        apply AxiomII in H9; destruct H9, H10, H10; add (X≠∅) H.
        apply LemmaZorn_2 with (A:=En_A X B le)(y:=y)(x:=u) in H; auto.
        apply AxiomII; split; auto; exists x; split; auto.
        apply AxiomII; split; Ens; rewrite <- H10; auto.
Qed.

Hint Resolve Zorn : Axiom_of_Choice.

End Zorn_Lemma.
```

4.7 良序定理

定理 (良序定理)　*在每一个集合上都有一个良序.*

良序定理由 Zorn 引理可以证明, 其 Coq 描述及证明代码如下:

```
Require Export Zorn_Lemma.

Module Type WellOrder_Theorem.

Declare Module Zorn : Zorn_Lemma.

Definition En_L X :=
  \{\ λ Y le, Y ⊂ X /\ Y ≠ ∅ /\ WellOrder le Y\}\.

Definition lee X: Class :=
```

```
  \{\ λ L1 L2, (L1 ∈ (En_L X) /\ L2 ∈ (En_L X)) /\
  (∀ x y, x∈(First L1) /\ y∈(First L1) ->
  (Rrelation x (Second L1) y <-> Rrelation x (Second L2) y)) /\
  Initial_Segment (First L1) (First L2) (Second L2) \}\.

Definition En_Z K := \{ λ x, ∃ Y le, [Y,le] ∈ K /\ x∈Y \}.

Definition leeq K : Class :=
  \{\ λ u v, ∃ Y le, [Y,le] ∈ K /\ u∈Y /\ v∈Y /\ Rrelation u le v \}\.

Definition leep Y le x :=
  \{\ λ y1 y2, (y1∈Y /\ y2∈Y /\ Rrelation y1 le y2) \/
  (y1∈(Y∪[x]) /\ y2=x) \}\.

Definition leq Y := \{\ λ y1 y2, y1∈Y /\ y2∈Y /\ y1 = y2 \}\.

Lemma Property_lee : ∀ Y1 Y2 le1 le2,
  Ensemble ([[Y1, le1],[Y2, le2]]) -> First ([Y1,le1]) = Y1 /\
  Second ([Y1,le1]) = le1 /\ First ([Y2,le2]) = Y2 /\
  Second ([Y2,le2]) = le2.
Proof.
  intros.
  apply Theorem49 in H; destruct H.
  apply Theorem49 in H; apply Theorem54 in H; destruct H.
  apply Theorem49 in H0; apply Theorem54 in H0; destruct H0.
  repeat split; auto.
Qed.

Lemma Lemma_WP1 : ∀ X, Ensemble X -> PartialOrder (lee X) (En_L X).
Proof.
  intros.
  unfold PartialOrder; repeat split.
  - assert (Ensemble (pow(X)× pow(X×X))).
    { double H; apply Theorem38 in H0; auto; apply Theorem74.
      split; auto; apply Theorem38; auto; apply Theorem74; auto. }
    apply Theorem33 with (x:= pow(X)× pow(X×X)); auto.
    unfold Included; intros; PP H1 Y0 le0; apply AxiomII_P in H2.
    destruct H2, H3, H4; unfold WellOrder in H5; destruct H5; clear H6.
    unfold TotalOrder in H5; destruct H5; clear H6.
    unfold PartialOrder in H5; destruct H5, H6, H6.
    unfold Cartesian; apply AxiomII_P; repeat split; auto.
    + unfold PowerSet; apply AxiomII; split; auto.
    + unfold PowerSet; apply AxiomII; split; auto.
      * apply Theorem33 in H8; auto; apply Theorem74; auto.
      * unfold Included; intros; apply H8 in H9; PP H9 u v.
```

```
        apply AxiomII_P in H10; destruct H10, H11.
        apply AxiomII_P; repeat split; auto.
  - unfold Relation; intros; PP H0 x y; Ens.
  - unfold Included; intros; PP H0 u v; apply AxiomII_P in H1.
    destruct H1, H2, H2; apply AxiomII_P; auto.
  - unfold Reflexivity; intros; double H0; unfold En_L in H1.
    PP H1 u v; unfold Rrelation, lee; apply AxiomII_P.
    assert (Ensemble ([u,v])). { Ens. }
    split; try apply Theorem49; auto; apply Theorem49 in H3.
    apply Theorem54 in H3; destruct H3; rewrite H3, H4; clear H3 H4.
    split; Ens; split; intros.
    + split; intros; auto.
    + unfold Initial_Segment; split; try (unfold Included; auto).
      apply AxiomII_P in H2; destruct H2, H3, H4.
      split; try apply H5; intros; apply H6.
  - unfold Antisymmetry; intros; destruct H1.
    unfold Rrelation, lee in H1, H2; apply AxiomII_P in H1.
    apply AxiomII_P in H2; destruct H1, H2, H3, H4; clear H2 H3 H4.
    destruct H0; PP H0 Y1 le1; PP H2 Y2 le2; clear H0 H2 H3 H4.
    double H1; apply Property_lee in H1; destruct H1, H2, H3.
    rewrite H1, H2, H3, H4 in H5, H6; clear H1 H2 H3 H4.
    destruct H5, H6, H2, H4, H5, H6; apply Theorem49 in H0.
    destruct H0; clear H9; apply Theorem49 in H0.
    apply Theorem55; auto; split. 1: apply Theorem27; auto.
    clear H7 H8; unfold WellOrder in H5, H6; destruct H5, H6.
    clear H7 H8; unfold TotalOrder in H5, H6; destruct H5, H6.
    clear H7 H8; unfold PartialOrder in H5, H6; destruct H5, H6.
    clear H5 H6; destruct H7, H8; clear H6 H8; destruct H5, H7.
    clear x y; apply AxiomI; split; intros.
    + double H9; unfold Included in H8; apply H8 in H10.
      PP H10 x y; apply AxiomII_P in H11; destruct H11.
      apply H1 in H12; unfold Rrelation in H12; apply H12; auto.
    + double H9; unfold Included in H8; apply H6 in H10.
      PP H10 x y; apply AxiomII_P in H11; destruct H11.
      apply H3 in H12; unfold Rrelation in H12; apply H12; auto.
  - clear H; unfold Transitivity; intros; destruct H, H0.
    elim H; intros; destruct H3; unfold En_L in H2, H3, H4.
    PP H2 Y1 le1;PP H3 Y2 le2;PP H4 Y3 le3;clear H H2 H3 H4 H5 H6 H7.
    unfold Rrelation; unfold lee; apply AxiomII_P.
    unfold Rrelation in H0, H1; unfold lee in H0, H1.
    apply AxiomII_P in H0; apply AxiomII_P in H1;
    destruct H0, H1; split.
    + apply Theorem49 in H; apply Theorem49 in H1; destruct H, H1.
      apply Theorem49; split; auto.
    + apply Property_lee in H; apply Property_lee in H1.
```

```
      destruct H, H1, H3, H4, H5, H6; clear H1 H4.
      rewrite H, H3, H5, H7 in H0; rewrite H5, H6, H7, H8 in H2.
      rewrite H, H3, H6, H8; clear H H3 H5 H7 H6 H8; destruct H0, H2.
      destruct H, H1; split; auto; clear H H1 H3 H4; destruct H0, H2.
      unfold Initial_Segment in H0, H2; destruct H0, H2, H3, H4; split;
      intros.
      * elim H7; intros; unfold Included in H0; apply H0 in H8.
        apply H0 in H9; add (y ∈ Y2) H8; clear H9; apply H1 in H8.
        apply H in H7; split; intros; tauto.
      * double H0; add (Y2 ⊂ Y3) H7; apply Theorem28 in H7.
        unfold Initial_Segment; repeat split;
        try apply H4; auto; intros.
        apply H5 with (v:=v0); destruct H8, H9; assert (u0 ∈ Y2).
        { apply H6 with (v:=v0); unfold Included in H0.
          apply H0 in H9; repeat split; auto. }
        repeat split; auto; unfold Included in H0; apply H0 in H9.
        add (v0 ∈ Y2) H11; apply H1 in H11; apply H11; auto.
Qed.

Lemma Lemma_WP2 : ∀ (X K : Class),
  Ensemble X -> Chain K (En_L X)(lee X) -> WellOrder(leeq K)(En_Z K).
Proof.
  intros; double H.
  unfold Chain in H0; apply (Lemma_WP1 X) in H1; apply H0 in H1.
  clear H0; destruct H1; unfold WellOrder; split.
  - unfold TotalOrder; split; intros.
    { unfold PartialOrder; repeat split.
      - assert ((En_Z K) ⊂ X).
        { unfold Included; intros; unfold En_Z in H2.
          apply AxiomII in H2; destruct H2, H3 as [Y [le H3]], H3.
          destruct H0; unfold Included in H0; apply H0 in H3.
          unfold En_L in H3; apply AxiomII_P in H3; destruct H3, H6.
          unfold Included in H6; apply H6 in H4; auto. }
        apply Theorem33 in H2; auto.
      - unfold Relation; intros; PP H2 Y0 le0; Ens.
      - unfold Included; intros; PP H2 Y0 le0; apply AxiomII_P in H3.
        destruct H3, H4 as [Y1 [le1 H4]], H4, H5, H6; apply AxiomII_P.
        repeat split; try (apply AxiomII; split); Ens.
      - unfold Reflexivity; intros; unfold En_Z in H2;
        apply AxiomII in H2.
        unfold Rrelation, leeq; apply AxiomII_P; destruct H2.
        destruct H3 as [Y [le H3]],H3;split;try apply Theorem49;auto.
        exists Y, le; repeat split; auto; unfold Included in H0.
        apply H0 in H3; unfold En_L in H3; apply AxiomII_P in H3.
        destruct H3, H5; unfold WellOrder in H6; destruct H6; clear H6.
```

```
    destruct H7; clear H7; unfold TotalOrder in H6;
    destruct H6; clear H7.
    unfold PartialOrder in H6; destruct H6, H7, H8; clear H9.
    unfold Reflexivity in H8; apply H8; auto.
  - unfold Antisymmetry; intros; destruct H3; apply AxiomII_P in H3.
    apply AxiomII_P in H4; destruct H3, H4; clear H4.
    destruct H5 as [Y1 [le1 H5]], H6 as [Y2 [le2 H6]].
    destruct H5, H6, H5, H7, H8, H9.
    generalize (classic ([Y1, le1] = [Y2, le2])); intros;
    destruct H12.
    + assert (Ensemble ([Y1,le1])). { Ens. }
      apply Theorem49 in H13; apply Theorem55 in H12; auto.
      clear H13; destruct H12; rewrite H13 in H10; clear H12.
      clear H13; destruct H0; clear H12; unfold Included in H0.
      apply H0 in H6; apply AxiomII_P in H6; destruct H6, H12, H13.
      unfold WellOrder in H14; destruct H14; clear H15.
      unfold TotalOrder in H14; destruct H14; clear H15.
      unfold PartialOrder in H14; destruct H14, H15, H16, H17.
      unfold Antisymmetry in H17; apply H17; auto.
    + assert ([Y1, le1] ∈ K /\ [Y2, le2] ∈ K); Ens.
      clear H2; unfold TotalOrder in H1; destruct H1.
      apply H2 in H13; auto; clear H2 H12; destruct H13.
      * unfold Rrelation in H2; apply Theorem4' in H2; destruct H2.
        clear H12; unfold lee in H2; apply AxiomII_P in H2;
        destruct H2.
        apply Property_lee in H2; destruct H2, H13, H14.
        rewrite H2, H13, H14, H15 in H12; clear H2 H13 H14 H15.
        destruct H12, H12; apply H12 in H10; auto; clear H2 H12 H13.
        destruct H0; unfold Included in H0; apply H0 in H6.
        apply AxiomII_P in H6; destruct H6, H12, H13.
        unfold WellOrder in H14; destruct H14; clear H15.
        unfold TotalOrder in H14; destruct H14; clear H15.
        unfold PartialOrder in H14; destruct H14, H15, H16, H17.
        unfold Antisymmetry in H17; apply H17; auto.
      * unfold Rrelation in H2; apply Theorem4' in H2; destruct H2.
        clear H12; unfold lee in H2; apply AxiomII_P in H2;
        destruct H2.
        apply Property_lee in H2; destruct H2, H13, H14.
        rewrite H2, H13, H14, H15 in H12; clear H2 H13 H14 H15.
        destruct H12, H12; apply H12 in H10; auto; clear H2 H12 H13.
        destruct H0; unfold Included in H0; apply H0 in H6.
        apply AxiomII_P in H6; destruct H6, H12, H13.
        unfold WellOrder in H14; destruct H14; clear H15.
        unfold TotalOrder in H14; destruct H14; clear H15.
        unfold PartialOrder in H14; destruct H14, H15, H16, H17.
```

```
      unfold Antisymmetry in H17; apply H17; auto.
  - unfold Transitivity;intros;destruct H2;clear H2;destruct H3.
    unfold Rrelation, leeq in H2, H3; apply AxiomII_P in H2.
    apply AxiomII_P in H3; destruct H2, H3;
    destruct H4 as [Y1 [le1 H4]].
    destruct H5 as [Y2 [le2 H5]], H4, H5, H6, H7, H8, H9.
    generalize (classic ([Y1, le1] = [Y2, le2])); intros;
    destruct H12.
    + assert (Ensemble ([Y1, le1])); Ens.
      apply Theorem49 in H13; apply Theorem55 in H12; auto.
      clear H13; destruct H12; rewrite H12 in H6;
      rewrite H13 in H10.
      unfold Rrelation, leeq; apply AxiomII_P; split.
      * apply Theorem49; Ens.
      * exists Y2, le2; repeat split; auto.
        unfold Included in H0; apply H0 in H5; unfold En_L in H5.
        apply AxiomII_P in H5; destruct H5, H14, H15 as [H16 H15].
        clear H16; unfold WellOrder in H15; destruct H15; clear H16.
        unfold TotalOrder in H15; destruct H15; clear H16.
        unfold PartialOrder in H15; destruct H15, H16, H17, H18.
        unfold Transitivity in H19; apply H19 with (v:=v); auto.
    + unfold TotalOrder in H1; destruct H1; apply H13 in H12; auto.
      clear H13; destruct H12.
      * unfold Rrelation, lee in H12; apply Theorem4' in H12;
        destruct H12.
        clear H13; apply AxiomII_P in H12; destruct H12.
        apply Property_lee in H12; destruct H12, H14, H15.
        rewrite H12, H14, H15, H16 in H13; clear H12 H14 H15 H16.
        destruct H13, H13, H14; clear H15; unfold Rrelation, leeq.
        apply AxiomII_P; split; try (apply Theorem49; Ens).
        exists Y2, le2; repeat split; auto; double H6.
        add (v∈Y1) H15; apply H13 in H15; apply H15 in H10;
        clear H13 H15.
        unfold Included in H0; apply H0 in H5; unfold En_L in H5.
        apply AxiomII_P in H5; destruct H5, H13, H15 as [H16 H15].
        clear H16; unfold WellOrder in H15; destruct H15; clear H16.
        unfold TotalOrder in H15; destruct H15; clear H16.
        unfold PartialOrder in H15; destruct H15, H16, H17, H18.
        unfold Transitivity in H19; apply H19 with (v:=v); auto.
      * unfold Rrelation, lee in H12; apply Theorem4' in H12;
        destruct H12.
        clear H13; apply AxiomII_P in H12; destruct H12.
        apply Property_lee in H12; destruct H12, H14, H15.
        rewrite H12, H14, H15, H16 in H13; clear H12 H14 H15 H16.
        destruct H13, H13, H14; clear H15; unfold Rrelation, leeq.
```

```
      apply AxiomII_P; split; try (apply Theorem49; Ens).
      exists Y1, le1; repeat split; auto; double H7.
      add (w∈Y2) H15; apply H13 in H15; apply H15 in H11;
      clear H13 H15.
      unfold Included in H0; apply H0 in H4; unfold En_L in H4.
      apply AxiomII_P in H4; destruct H4, H13, H15 as [H16 H15].
      clear H16; unfold WellOrder in H15; destruct H15; clear H16.
      unfold TotalOrder in H15; destruct H15; clear H16.
      unfold PartialOrder in H15; destruct H15, H16, H17, H18.
      unfold Transitivity in H19; apply H19 with (v:=v); auto. }
{ unfold En_Z in H2; destruct H2; apply AxiomII in H2;
  apply AxiomII in H4.
  destruct H2,H4,H5 as [Y1 [le1 H5]],H6 as [Y2 [le2 H6]], H5, H6.
  generalize (classic([Y1,le1] = [Y2,le2]));intros;destruct H9.
  - assert (Ensemble ([Y1, le1])). { Ens. }
    apply Theorem49 in H10; apply Theorem55 in H9; auto.
    clear H10; destruct H9; rewrite H9 in H7;
    clear H9 H10; double H6.
    unfold Included in H0; apply H0 in H9; unfold En_L in H9.
    apply AxiomII_P in H9; destruct H9, H10; clear H9; clear H10.
    destruct H11 as [H12 H11]; clear H12; unfold WellOrder in H11.
    destruct H11; clear H10; unfold TotalOrder in H9; destruct H9.
    clear H9; double H7; add (y∈Y2) H9; apply H10 in H9; auto.
    clear H10 H3; destruct H9.
    + left; unfold Rrelation, leeq; apply AxiomII_P.
      split; try apply Theorem49; Ens.
      exists Y2, le2; repeat split; auto.
    + right; unfold Rrelation, leeq; apply AxiomII_P.
      split; try apply Theorem49; Ens.
      exists Y2, le2; repeat split; auto.
  - unfold TotalOrder in H1; destruct H1; apply H10 in H9; auto.
    clear H10; destruct H9.
    + unfold Rrelation, lee in H9; apply Theorem4' in H9;
      destruct H9.
      clear H10; apply AxiomII_P in H9; destruct H9.
      apply Property_lee in H9; destruct H9, H11, H12.
      rewrite H9, H11, H12, H13 in H10; clear H9 H11 H12 H13.
      destruct H10; clear H9; unfold Initial_Segment in H10;
      destruct H10, H10, H11.
      clear H12; unfold WellOrder in H11; destruct H11.
      clear H12; unfold TotalOrder in H11; destruct H11.
      unfold Included in H10; apply H10 in H7; double H7.
      add (y∈Y2) H13; apply H12 in H13; auto; clear H12;
      destruct H13.
      * left; unfold Rrelation, leeq; apply AxiomII_P.
```

```
        split; try apply Theorem49; Ens.
        exists Y2, le2; repeat split; auto.
      * right; unfold Rrelation, leeq; apply AxiomII_P.
        split; try apply Theorem49; Ens.
        exists Y2, le2; repeat split; auto.
    + unfold Rrelation, lee in H9; apply Theorem4' in H9;
      destruct H9.
      clear H10; apply AxiomII_P in H9; destruct H9.
      apply Property_lee in H9; destruct H9, H11, H12.
      rewrite H9, H11, H12, H13 in H10; clear H9 H11 H12 H13.
      destruct H10; clear H9; unfold Initial_Segment in H10;
      destruct H10, H10, H11.
      clear H12; unfold WellOrder in H11; destruct H11.
      clear H12; unfold TotalOrder in H11; destruct H11.
      unfold Included in H10; apply H10 in H8; double H7.
      add (y∈Y1) H13; apply H12 in H13;
      auto; clear H12; destruct H13.
      * left; unfold Rrelation, leeq; apply AxiomII_P.
        split; try apply Theorem49; Ens.
        exists Y1, le1; repeat split; auto.
      * right; unfold Rrelation, leeq; apply AxiomII_P.
        split; try apply Theorem49; Ens.
        exists Y1, le1; repeat split; auto. }
- intro P; intros; destruct H2; apply Property_NotEmpty in H3.
  destruct H3 as [p H3]; double H3; unfold Included in H2.
  apply H2 in H4; clear H2; unfold En_Z in H4; apply AxiomII in H4.
  destruct H4, H4 as [Y [le H4]], H4; clear H2; double H4.
  apply H0 in H4; unfold En_L in H4; apply AxiomII_P in H4.
  destruct H4, H6; clear H4; clear H6; unfold WellOrder in H7.
  destruct H7; clear H4; assert ((Y ∩ P) ⊂ Y /\ (Y ∩ P) ≠ ∅).
  { split.
    - unfold Included; intros; apply Theorem4' in H4; apply H4.
    - apply Property_NotEmpty; exists p; apply Theorem4'; auto. }
  clear H3; clear H5; elim H4; intros; apply H6 in H4; clear H6.
  clear H3; destruct H4 as [q H3]; unfold MinElement in H3.
  apply H3 in H5; clear H3; destruct H5; apply Theorem4' in H3.
  destruct H3; exists q; unfold MinElement; split; auto; clear H6.
  intro r; intros; intro; unfold Rrelation in H7; destruct H7.
  unfold leeq in H7; apply AxiomII_P in H7; destruct H7.
  clear H7; destruct H9 as [Y1 [le1 H7]], H7, H9, H10.
  unfold TotalOrder in H1; destruct H1; unfold Connect in H11.
  generalize (classic ([Y,le] = [Y1,le1])); intros; destruct H13.
  + assert (Ensemble ([Y, le])); Ens.
    apply Theorem49 in H14; apply Theorem55 in H13; auto.
    clear H14; destruct H13; rewrite <- H13 in H9; rewrite H14 in H4.
```

```
      apply H4 with (y:=r); try apply Theorem4'; auto.
    + apply H12 in H13; auto; clear H12; destruct H13.
      * unfold Rrelation in H12; apply Theorem4' in H12; destruct H12.
        clear H13; unfold lee in H12; apply AxiomII_P in H12.
        destruct H12; apply Property_lee in H12; destruct H12,H14,H15.
        rewrite H12, H14, H15, H16 in H13; clear H12 H14 H15 H16.
        destruct H13; unfold Initial_Segment in H13; destruct H13, H14.
        assert (r∈Y1 /\ q∈Y /\ Rrelation r le1 q).
        { repeat split; auto. } apply H15 in H16; clear H15.
        apply H4 with (y:=r); try apply Theorem4'; auto.
        split; auto; apply H13; auto.
      * unfold Rrelation in H12; apply Theorem4' in H12; destruct H12.
        clear H13; unfold lee in H12; apply AxiomII_P in H12.
        destruct H12; apply Property_lee in H12; destruct H12,H14,H15.
        rewrite H12, H14, H15, H16 in H13; clear H12 H14 H15 H16.
        destruct H13; unfold Initial_Segment in H13; destruct H13, H14.
        clear H15; double H9; unfold Included in H14; apply H14 in H15.
        apply H4 with (y:=r); try apply Theorem4'; auto.
        split; auto; apply H13; auto.
Qed.

Lemma Lemma_WP3 : ∀ (K X: Class),
  Ensemble X -> Chain K (En_L X) (lee X) ->
  ∃ y, BoundU y K (En_L X) (lee X).
Proof.
  intros; double H; double H0.
  apply Lemma_WP2 in H0; auto; exists ([(En_Z K),(leeq K)]).
  unfold Chain in H2; apply (Lemma_WP1 X) in H1; apply H2 in H1.
  clear H2; destruct H1; unfold BoundU; intros; destruct H3.
  assert ([En_Z K, leeq K] ∈ (En_L X)).
  { unfold En_L; apply AxiomII_P; split.
    - apply Theorem49; split.
      + assert ((En_Z K) ⊂ X).
        { unfold Included; intros; unfold En_Z in H5.
          apply AxiomII in H5; destruct H5, H6 as [Y [le H6]], H6.
          unfold Included in H1; apply H1 in H6; unfold En_L in H6.
          apply AxiomII_P in H6; destruct H6, H8; auto. }
        apply Theorem33 in H5; auto.
      + assert ((leeq K) ⊂ X × X ).
        { unfold Included; intros; unfold leeq in H5.
          PP H5 u v; apply AxiomII_P in H6; destruct H6.
          destruct H7 as [Y [le H7]], H7, H8, H9.
          unfold Included in H1; apply H1 in H7; unfold En_L in H7.
          apply AxiomII_P in H7; destruct H7, H11;
          unfold Included in H11.
```

```
        apply H11 in H8; apply H11 in H9; unfold Cartesian.
        apply AxiomII_P; repeat split; auto. }
      assert (Ensemble X /\ Ensemble X). { auto. }
      apply Theorem74 in H6; apply Theorem33 in H5; auto.
  - split.
    + unfold Included; intros; unfold En_Z in H5.
      apply AxiomII in H5; destruct H5, H6 as [Y [le H6]], H6.
      unfold Included in H1; apply H1 in H6; unfold En_L in H6.
      apply AxiomII_P in H6; destruct H6, H8; auto.
    + split; try apply H0; destruct H1; apply Property_NotEmpty in H5.
      destruct H5; apply Property_NotEmpty; double H5.
      unfold Included in H1; apply H1 in H6; PP H6 Y0 le0.
      apply AxiomII_P in H7; destruct H7, H8, H9.
      apply Property_NotEmpty in H9; destruct H9.
      exists x0; unfold En_Z; apply AxiomII; split; Ens. }
repeat split; auto; try apply H1; intros.
double H6; unfold Included in H1; apply H1 in H7.
double H7; unfold En_L in H8; PP H8 Y1 le1; clear H9.
unfold Rrelation; unfold lee; apply AxiomII_P.
assert (Ensemble ([Y1,le1]) /\ Ensemble ([En_Z K,leeq K])). { Ens. }
split; try apply Theorem49; auto; destruct H9.
apply Theorem49 in H9; apply Theorem54 in H9; destruct H9.
apply Theorem49 in H10; apply Theorem54 in H10; destruct H10.
rewrite H9, H10, H11, H12; clear H9; clear H10; clear H11; clear H12.
clear H3; clear H4; split; auto; split; intros.
- destruct H3; split; intros.
  + unfold Rrelation, leeq; apply AxiomII_P.
    split; try apply Theorem49; Ens.
    exists Y1, le1; repeat split; auto.
  + unfold Rrelation, leeq in H9; apply AxiomII_P in H9.
    destruct H9, H10 as [Y2 [le2 H10]], H10, H11, H12, H2.
    generalize (classic ([Y1,le1] = [Y2,le2])); intros; destruct H15.
    * assert (Ensemble ([Y1, le1])). { Ens. }
      apply Theorem49 in H16; apply Theorem55 in H15; auto.
      clear H16; destruct H15; rewrite H16; auto.
    * apply H14 in H15; auto; clear H14; destruct H15.
      -- unfold Rrelation in H14; apply Theorem4' in H14; destruct H14.
         clear H15; unfold lee in H14; apply AxiomII_P in H14.
         destruct H14; apply Property_lee in H14;
         destruct H14, H16, H17.
         rewrite H14, H16, H17, H18 in H15; clear H14; clear H16.
         clear H17; clear H18; destruct H15, H15; apply H15; auto.
      -- unfold Rrelation in H14; apply Theorem4' in H14; destruct H14.
         clear H15; unfold lee in H14; apply AxiomII_P in H14.
         destruct H14; apply Property_lee in H14;
```

```
          destruct H14, H16, H17.
          rewrite H14, H16, H17, H18 in H15; clear H14; clear H16.
          clear H17; clear H18; destruct H15, H15; apply H15; auto.
  - unfold Initial_Segment; split.
    + unfold Included; intros; apply AxiomII; split; Ens.
    + split; try apply H0; intros; destruct H3, H4.
      unfold Rrelation, leeq in H9; apply AxiomII_P in H9.
      destruct H9, H10 as [Y2 [le2 H10]], H10, H11, H12, H2.
      generalize (classic ([Y1,le1] = [Y2,le2])); intros; destruct H15.
      * assert (Ensemble ([Y1, le1])). { Ens. }
        apply Theorem49 in H16; apply Theorem55 in H15; auto.
        clear H16; destruct H15; rewrite H15; auto.
      * apply H14 in H15; auto; clear H14; destruct H15.
        -- unfold Rrelation in H14; apply Theorem4' in H14; destruct H14.
           clear H15; unfold lee in H14; apply AxiomII_P in H14.
           destruct H14; apply Property_lee in H14;
           destruct H14, H16, H17.
           rewrite H14, H16, H17, H18 in H15; clear H14 H16 H17 H18.
           destruct H15; clear H14; unfold Initial_Segment in H15.
           destruct H15, H15, H16; apply H17 with (v:=v); auto.
        -- unfold Rrelation in H14; apply Theorem4' in H14; destruct H14.
           clear H15; unfold lee in H14; apply AxiomII_P in H14.
           destruct H14; apply Property_lee in H14;
           destruct H14, H16, H17.
           rewrite H14, H16, H17, H18 in H15; clear H14 H16 H17 H18.
           destruct H15; clear H14; unfold Initial_Segment in H15.
           destruct H15, H15; auto.
Qed.

Theorem WellOrderTheorem : ∀ (X: Class),
  Ensemble X -> ∃ le0: Class, WellOrder le0 X.
Proof.
  intros.
  assert (PartialOrderSet (En_L X) (lee X)).
  { unfold PartialOrderSet; try apply Lemma_WP1; auto. }
  double H0; apply Zorn.Zorn in H1; intros; try apply Lemma_WP3; auto.
  destruct H1 as [Y H1]; unfold MaxElement in H1.
  generalize (classic (X = ∅)); intros; destruct H2.
  { rewrite H2; exists ∅; unfold WellOrder; split; intros.
    - unfold TotalOrder; split; intros.
      + unfold PartialOrder; repeat split; intros.
        * apply Theorem33 with (x:=X); auto; unfold Included; intros.
          generalize (Theorem16 z); intros; contradiction.
        * unfold Relation; intros.
          generalize (Theorem16 z); intros; contradiction.
```

```
      * unfold Included; intros.
        generalize (Theorem16 z); intros; contradiction.
      * unfold Reflexivity; intros.
        generalize (Theorem16 a); intros; contradiction.
      * unfold Antisymmetry; intros; destruct H3.
        generalize (Theorem16 x); intros; contradiction.
      * unfold Transitivity; intros; destruct H3, H3.
        generalize (Theorem16 u); intros; contradiction.
    + destruct H3; generalize (Theorem16 x); contradiction.
  - destruct H3; generalize (Theorem26 A); intros.
    absurd (A = ∅); auto; apply Theorem27; auto. }
assert (En_L X ≠ ∅).
{ apply Property_NotEmpty in H2; destruct H2.
  apply Property_NotEmpty; exists ([[x],leq ([x])]).
  unfold En_L; apply AxiomII_P; repeat split; intros.
  - assert (Ensemble ([x])). { apply Theorem42; Ens. }
    apply Theorem49; split; auto.
    apply Theorem33 with (x:= ([x])×([x])); auto.
    + apply Theorem74; auto.
    + unfold Included; intros; PP H4 a b; unfold leq in H5.
      apply AxiomII_P in H5; destruct H5, H6, H7.
      unfold Cartesian; apply AxiomII_P; repeat split; auto.
  - unfold Included; intros; unfold Singleton in H3.
    apply AxiomII in H3; destruct H3; rewrite H4; auto.
    apply Theorem19; Ens.
  - apply Property_NotEmpty; exists x; apply AxiomII; Ens.
  - apply Theorem33 with (x:=X); auto; unfold Included; intros.
    unfold Singleton in H3; apply AxiomII in H3; destruct H3.
    rewrite H4; auto; apply Theorem19; Ens.
  - unfold Relation; intros; PP H3 a b; exists a, b; auto.
  - unfold Included; intros; PP H3 a b; unfold leq in H4.
    apply AxiomII_P in H4; destruct H4, H5, H6.
    unfold Cartesian; apply AxiomII_P; repeat split; auto.
  - unfold Reflexivity; intros; unfold Rrelation, leq.
    apply AxiomII_P; repeat split; auto; apply Theorem49; Ens.
  - unfold Antisymmetry; intros; destruct H3.
    unfold Singleton in H3, H5; apply AxiomII in H3.
    apply AxiomII in H5; destruct H3, H5.
    rewrite H6, H7; try apply Theorem19; Ens.
  - unfold Transitivity; intros; destruct H3, H3, H5.
    apply AxiomII in H3; apply AxiomII in H6; destruct H3, H6.
    rewrite H7, H8; try apply Theorem19; Ens.
    unfold Rrelation, leq; apply AxiomII_P; repeat split; auto.
    + apply Theorem49; split; Ens.
    + unfold Singleton; apply AxiomII; Ens.
```

```
    + unfold Singleton; apply AxiomII; Ens.
  - destruct H3; apply AxiomII in H3; apply AxiomII in H5.
    destruct H3, H5; rewrite H6, H7 in H4; try apply Theorem19; Ens.
    contradiction.
  - destruct H3; apply Property_NotEmpty in H4; destruct H4.
    exists x0; unfold MinElement; intros; split; auto; intros.
    intro; destruct H7; unfold Rrelation, leq in H7.
    apply AxiomII_P in H7; destruct H7, H9, H10.
    rewrite H11 in H8; contradiction. }
apply H1 in H3; clear H1 H2; destruct H3.
unfold En_L in H1; PP H1 Y0 le0; apply AxiomII_P in H3.
destruct H3,H4,H5;apply Property_ProperIncluded in H4;destruct H4.
- double H4; unfold ProperIncluded in H7; destruct H7; clear H8.
  apply Property_ProperIncluded' in H4; destruct H4, H4.
  assert (([Y0∪[x],(leep Y0 le0 x)]) ∈ (En_L X)).
  { unfold WellOrder in H6; destruct H6;
    unfold TotalOrder in H6; destruct H6.
    double H6; unfold PartialOrder in H11; destruct H11 as [H12 H11].
    clear H12; destruct H11; unfold En_L;
    apply AxiomII_P; repeat split; intros.
    - apply Theorem49 in H3; destruct H3; apply Theorem49; split.
      + apply AxiomIV; split; auto; apply Theorem42; Ens.
      + assert ((leep Y0 le0 x) ⊂ le0∪(X×X)).
        { unfold Included; intros; PP H14 y1 y2;
          apply AxiomII_P in H15.
          destruct H15; apply Theorem4; destruct H16.
          - left; destruct H16, H17; auto.
          - right; unfold Cartesian; apply AxiomII_P; destruct H16.
            rewrite H17; repeat split; try apply Theorem49; Ens.
            apply Theorem4 in H16; destruct H16; auto.
            unfold Singleton in H16; apply AxiomII in H16;
            destruct H16.
            rewrite H18; auto; apply Theorem19; Ens. }
        apply Theorem33 in H14; auto; apply AxiomIV; split; auto.
        apply Theorem74; auto.
    - unfold Included; intros; apply Theorem4 in H13; destruct H13.
      + unfold Included in H7; apply H7 in H13; auto.
      + unfold Singleton in H13; apply AxiomII in H13; destruct H13.
        rewrite H14; auto; apply Theorem19; Ens.
    - apply Property_NotEmpty in H5; destruct H5.
      apply Property_NotEmpty; exists x0; apply Theorem4; tauto.
    - apply Theorem33 with (x:=X); auto; unfold Included; intros.
      apply Theorem4 in H13; destruct H13; auto;
      unfold Singleton in H13.
      apply AxiomII in H13; destruct H13; rewrite H14; auto.
```

```
    apply Theorem19; Ens.
  - unfold Relation; intros; unfold leep in H13; PP H13 y1 y2; Ens.
  - unfold Included; intros; PP H13 y1 y2; apply AxiomII_P in H14.
    destruct H14; apply AxiomII_P; split; auto; destruct H15.
    + destruct H15, H16; split; try apply Theorem4; auto.
    + destruct H15; split; auto; rewrite H16; apply Theorem4.
      right; unfold Singleton; apply AxiomII; Ens.
  - unfold Reflexivity; intros; unfold Rrelation, leep;
    apply AxiomII_P.
    split; try apply Theorem49; Ens; apply Theorem4 in H13;
    destruct H13.
    + left; repeat split; auto; destruct H12;
      unfold Reflexivity in H12; auto.
    + right; unfold Singleton in H13;
      apply AxiomII in H13; destruct H13.
      rewrite H14; try apply Theorem19; Ens; split; auto.
      apply Theorem4; right; apply AxiomII; Ens.
  - unfold Antisymmetry; intros; clear H13; destruct H14.
    unfold Rrelation, leep in H13, H14; apply AxiomII_P in H13.
    apply AxiomII_P in H14; destruct H13, H14, H15, H16.
    + destruct H15, H16, H17, H18, H12, H21; clear H22.
      unfold Antisymmetry in H21; apply H21; auto.
    + destruct H15, H16; rewrite H18 in H15; contradiction.
    + destruct H15, H16; rewrite H17 in H16; contradiction.
    + destruct H15, H16; rewrite H17, H18; auto.
  - clear H11; destruct H12; clear H11; destruct H12;
    unfold Transitivity.
    intros;unfold Transitivity in H12;destruct H13,H14,H13,H16.
    unfold Rrelation; apply AxiomII_P; split;
    try apply Theorem49; Ens.
    unfold Rrelation, leep in H14, H15; apply AxiomII_P in H14.
    apply AxiomII_P in H15; destruct H14, H15, H18, H19.
    + left; destruct H18, H19, H20, H21; repeat split; auto.
      apply H12 with (v:=v); auto.
    + right; destruct H19; split; auto.
    + destruct H18, H19; rewrite H20 in H19; contradiction.
    + right; destruct H19; split; auto.
  - destruct H13; apply Theorem4 in H13.
    apply Theorem4 in H15; destruct H13, H15.
    + double H13; add (y∈Y0) H16; apply H10 in H16; auto.
      clear H10; destruct H16.
      * left; unfold Rrelation, leep; apply AxiomII_P.
        repeat split; try apply Theorem49; Ens.
      * right; unfold Rrelation, leep; apply AxiomII_P.
        repeat split; try apply Theorem49; Ens.
```

```
  + left; unfold Rrelation, leep; apply AxiomII_P.
    split; try apply Theorem49; Ens; right; split.
    * apply Theorem4; tauto.
    * apply AxiomII in H15; destruct H15; apply H16.
      apply Theorem19; Ens.
  + right; unfold Rrelation, leep; apply AxiomII_P.
    split; try apply Theorem49; Ens; right; split.
    * apply Theorem4; tauto.
    * apply AxiomII in H13; destruct H13; apply H16.
      apply Theorem19; Ens.
  + left; unfold Rrelation, leep; apply AxiomII_P.
    split; try apply Theorem49; Ens; right; split.
    * apply Theorem4; tauto.
    * apply AxiomII in H15; destruct H15; apply H16.
      apply Theorem19; Ens.
- destruct H13; assert (A ⊂ Y0 \/ (∃ B, B⊂Y0 /\ A = B∪[x])).
  { generalize (classic (x ∈ A)); intros; destruct H15.
    - right; exists (A ~ [x]); split.
      + unfold Included; intros; unfold Setminus in H16.
        apply AxiomII in H16; destruct H16, H17.
        unfold Complement in H18; apply AxiomII in H18;
        destruct H18.
        unfold NotIn in H19; unfold Included in H13;
        apply H13 in H17.
        apply Theorem4 in H17; tauto.
      + unfold Setminus; apply AxiomI; split; intros.
        * unfold Included in H13; double H16; apply H13 in H17.
          apply Theorem4; apply Theorem4 in H17;
          destruct H17; try tauto.
          left; apply Theorem4'; split; auto; unfold Complement.
          apply AxiomII; split; Ens; unfold NotIn; intro.
          unfold Singleton in H18; apply AxiomII in H18;
          destruct H18.
          rewrite H19 in H17; try contradiction;
          try apply Theorem19; Ens.
        * apply Theorem4 in H16; destruct H16.
          -- apply Theorem4' in H16; apply H16.
          -- apply AxiomII in H16;destruct H16;rewrite H17;auto.
             apply Theorem19; Ens.
    - left; unfold Included; intros; unfold Included in H13.
      double H16; apply H13 in H17; apply Theorem4 in H17.
      destruct H17; auto; apply AxiomII in H17; destruct H17.
      rewrite H18 in H16; try contradiction;
      apply Theorem19; Ens. }
  destruct H15.
```

```
    + double H15; add (A≠∅) H16; apply H9 in H16; clear H9.
      destruct H16; exists x0; unfold MinElement in H9.
      apply H9 in H14; clear H9; destruct H14;
      unfold MinElement; intros.
      split; auto; intros; apply H14 in H17; intro;
      elim H17; clear H17.
      destruct H18; split; auto; unfold Rrelation, leep in H17.
      apply AxiomII_P in H17; destruct H17, H19; try apply H19.
      destruct H19; rewrite H20 in H9; apply H15 in H9;
      contradiction.
    + destruct H15 as [B H15], H15.
      generalize (classic (B = ∅)); intros; destruct H17.
      * rewrite H17 in H16; rewrite Theorem17 in H16; clear H17.
        rewrite H16; exists x; unfold MinElement; intros;
        split; intros.
        -- unfold Singleton; apply AxiomII; split; Ens.
        -- unfold Singleton in H18; apply AxiomII in H18;
           destruct H18.
           intro;destruct H20;rewrite H19 in H21;try contradiction.
           apply Theorem19; Ens.
      * rewrite H16; double H15; add (B≠∅) H18; clear H16;
        apply H9 in H18.
        clear H9; destruct H18; exists x0; unfold MinElement in H9.
        apply H9 in H17; clear H9; destruct H17;
        unfold MinElement; intros.
        clear H17; split; try (apply Theorem4; tauto); intros.
        apply Theorem4 in H17; destruct H17.
        -- apply H16 in H17; intro; elim H17; clear H17.
           destruct H18;split;auto;unfold Rrelation, leep in H17.
           apply AxiomII_P in H17;destruct H17,H19;try apply H19.
           destruct H19; rewrite H20 in H9; apply H15 in H9;
           contradiction.
        -- intro; destruct H18;apply AxiomII in H17;destruct H17.
           rewrite H20 in H18, H19; try apply Theorem19; Ens.
           unfold Rrelation; apply AxiomII_P in H18;
           destruct H18, H21, H21.
           ++ destruct H22; contradiction.
           ++ contradiction. }
double H9; apply H2 in H10; elim H10; clear H10; split.
+ unfold Rrelation, lee; apply AxiomII_P.
  assert (Ensemble ([[Y0,le0],[Y0∪[x],leep Y0 le0 x]])).
  { apply Theorem49; Ens. } split; auto.
  apply Property_lee in H10; destruct H10, H11, H12.
  rewrite H10, H11, H12, H13; clear H10 H11 H12 H13.
  split; try (split; auto; apply AxiomII_P; Ens); split; intros.
```

```
      * split; intros.
        -- destruct H10; unfold Rrelation, leep; apply AxiomII_P.
           split; try apply Theorem49; Ens; try tauto.
        -- unfold Rrelation, leep in H11; apply AxiomII_P in H11.
           destruct H11; destruct H12, H12; try apply H13.
           destruct H10; rewrite H13 in H14; contradiction.
      * unfold Initial_Segment; split.
        -- unfold Included; intros; apply Theorem4; tauto.
        -- unfold En_L in H9; apply AxiomII_P in H9;
           destruct H9, H10, H11.
           split; try apply H12; intros; destruct H13, H14.
           unfold Rrelation, leep; apply AxiomII_P in H15;
           destruct H15.
           destruct H16; try apply H16; destruct H16.
           rewrite H17 in H14; contradiction.
    + intro; apply Theorem49 in H3; apply Theorem55 in H10; auto.
      destruct H10; elim H8; rewrite H10; apply Theorem4; right.
      apply AxiomII; split; Ens.
  - rewrite H4 in H6; exists le0; auto.
Qed.

Hint Resolve WellOrderTheorem : Axiom_of_Choice.

End WellOrder_Theorem.
```

4.8　良序定理证明选择公理

本节将选择公理视为一个定理, 通过良序定理给出其证明, Coq 描述及代码如下:

```
Require Export Wellordering_Theorem.

Module Type AC_Proof.

Declare Module WellOrder : WellOrder_Theorem.

Definition En_CF X le :=
  \{\ λ x y, x ∈ (pow(X) ~ [∅]) /\ y ∈ x /\
  (∃ z0, MinElement z0 x le /\ y = z0) \}\.

Theorem WellOrder_Choice : ∀ X, Ensemble X ->
  ∃ c, Choice_Function c X.
```

```
Proof.
  intros.
  generalize (classic (X = ∅)); intros; destruct H0.
  (* X = ∅ *)
  - rewrite H0; exists ∅; unfold Choice_Function;
    repeat split; intros.
    + unfold Relation; intros.
      generalize (Theorem16 z); contradiction.
    + destruct H1; generalize (Theorem16 ([x,y])); contradiction.
    + unfold Included; intros; unfold Range in H1.
      apply AxiomII in H1; destruct H1, H2.
      generalize (Theorem16 ([x,z])); contradiction.
    + apply AxiomI; split; intros.
      * unfold Domain in H1; apply AxiomII in H1; destruct H1, H2.
        generalize (Theorem16 ([z,x])); contradiction.
      * unfold Setminus in H1; apply Theorem4' in H1; destruct H1.
        unfold Complement in H2; apply AxiomII in H2; destruct H2.
        unfold PowerSet in H1; apply AxiomII in H1; destruct H1.
        add (∅ ⊂ z) H4; try (apply Theorem26); apply Theorem27 in H4.
        assert (z ∈ [∅]). { apply AxiomII; split; auto. }
        contradiction.
    + unfold Domain in H1; apply AxiomII in H1; destruct H1, H2.
      generalize (Theorem16 ([A,x])); contradiction.
  (* X ≠ ∅ *)
  - double H; apply WellOrder.WellOrderTheorem in H1.
    destruct H1 as [le H1]; unfold WellOrder in H1; destruct H1.
    exists (En_CF X le); unfold Choice_Function.
    assert (Function (En_CF X le)).
    { unfold Function; split; intros.
      - unfold Relation; intros; PP H3 x y; exists x, y; auto.
      - destruct H3; apply AxiomII_P in H3; apply AxiomII_P in H4.
        destruct H3, H4, H5, H6, H7, H8, H9, H10, H9, H10.
        unfold TotalOrder in H1; destruct H1.
        assert (y ∈ X /\ z ∈ X).
        { unfold Setminus in H5; apply Theorem4' in H5; destruct H5.
          apply AxiomII in H5; destruct H5; split; auto. }
        generalize (classic (y = z)); intros; destruct H15; auto.
        apply H13 in H14; auto; clear H15.
        assert (x ≠ ∅).
        { generalize (classic (x ≠ ∅)); intros; destruct H15; auto.
          apply NNPP in H15; rewrite H15 in H7.
          generalize (Theorem16 y); contradiction. }
        unfold MinElement in H9, H10; destruct H9, H10, H14; auto.
        + rewrite <- H12 in H17; apply H17 in H7.
          apply not_and_or in H7; destruct H7; try contradiction.
```

```
          apply NNPP in H7; symmetry; auto.
        + rewrite <- H11 in H16; apply H16 in H8.
          apply not_and_or in H8; destruct H8; try contradiction.
          apply NNPP in H8; symmetry; auto. }
    split; auto; repeat split; intros.
    + unfold Included; intros; apply AxiomII in H4.
      destruct H4, H5 as [y H5]; apply AxiomII_P in H5.
      destruct H5, H6, H7; clear H8; unfold Setminus in H6.
      apply Theorem4' in H6; destruct H6; clear H8.
      apply AxiomII in H6; destruct H6; auto.
    + apply AxiomI; intro A; split; intros.
      * apply AxiomII in H4; destruct H4, H5 as [y H5].
        apply AxiomII_P in H5; apply H5.
      * apply AxiomII; split; Ens; double H4.
        unfold Setminus in H5; apply Theorem4' in H5; destruct H5.
        apply AxiomII in H5; destruct H5; unfold Complement in H6.
        apply AxiomII in H6; destruct H6; clear H6; unfold NotIn in H8.
        assert (A ∈ [∅] <-> Ensemble A /\ (∅ ∈ 𝒰 -> A = ∅)).
        { split; intros.
          - unfold Singleton in H5; apply AxiomII in H6; auto.
          - unfold Singleton; apply AxiomII; auto. }
        apply Lemma_z in H6; auto; clear H8.
        apply not_and_or in H6; destruct H6; try contradiction.
        apply imply_to_and in H6; destruct H6; clear H6.
        assert (A ⊂ X /\ A ≠ ∅). { split; auto. }
        apply H2 in H6; destruct H6 as [z0 H6]; double H6.
        unfold MinElement in H9; apply H9 in H8; clear H9; destruct H8.
        exists z0; apply AxiomII_P; repeat split; auto.
        -- apply Theorem49; split; Ens.
        -- exists z0; auto.
    + double H4; apply Property_Value in H4; auto; unfold Domain in H5.
      apply AxiomII in H5; destruct H5, H6 as [y H6]; double H6.
      apply AxiomII_P in H7; destruct H7, H8, H9; clear H10.
      add ([A,y] ∈ (En_CF X le)) H4; unfold Function in H3.
      apply H3 in H4; rewrite H4; auto.
Qed.

Hint Resolve WellOrder_Choice : Axiom_of_Choice.

End AC_Proof.
```

4.9　Zermelo 假定证明选择公理

本节将选择公理视为一个定理, 通过 Zermelo 假定给出其证明, Coq 描述及代

码如下:

```
Require Export Zermelo_Postulate.

(** The proof of AC **)

Module Type Zermelo_Proof_AC.

Declare Module Ze : Zermelo_Postulate.

Definition En_p X : Class :=
  \{ λ z, ∃ A, A ∈ (pow(X) ~ [∅]) /\ z = (A × [A]) \}.

Theorem Zermelo_AC : ∀ X, Ensemble X ->
  ∃ c, Choice_Function c X.
Proof.
  intros.
  assert (∃ D, ∀ p, p ∈ (En_p X) -> ∃ x, p ∩ D = [x]).
  { assert (Ensemble (En_p X) /\ ∅ ∉ (En_p X)).
    { assert ((En_p X) ⊂ pow(X × (pow(X)~[∅]))).
      { unfold Included; intros; unfold En_p in H0; apply AxiomII in H0.
        destruct H0, H1 as [A H1], H1; rewrite H2 in *; clear H2.
        apply AxiomII; split; auto; unfold Included; intros.
        PP H2 a b; clear H2; apply AxiomII_P in H3; destruct H3, H3.
        unfold Setminus in H1; apply AxiomII in H1;
        destruct H1, H5; double H5.
        unfold PowerSet in H7; apply AxiomII in H7;
        destruct H7 as [_ H7].
        unfold Cartesian; apply AxiomII_P; repeat split; auto.
        unfold Singleton in H4; apply AxiomII in H4; destruct H4.
        rewrite H8 in *; try apply Theorem19; Ens; clear H8.
        unfold Setminus; apply Theorem4'; split; auto. }
      assert (Ensemble (pow( X × (pow(X) ~ [∅])))).
      { apply Theorem38; auto; apply Theorem74; split; auto.
        unfold Setminus; apply Theorem38 in H; auto.
        apply Theorem33 with (x:= pow(X)); auto.
        unfold Included; intros; apply Theorem4' in H1; apply H1. }
      apply Theorem33 in H0; auto; clear H1; split; auto.
      intro; unfold En_p in H1; apply AxiomII in H1; destruct H1.
      destruct H2 as [A H2], H2; unfold Setminus in H2.
      apply Theorem4' in H2; destruct H2; unfold Complement in H4.
      apply AxiomII in H4; destruct H4.
      generalize (classic (A = ∅)); intros; destruct H6.
      - rewrite H6 in H5; destruct H5; apply AxiomII; Ens.
      - apply Property_NotEmpty in H6; destruct H6.
```

```
      assert ([x,A] ∈ ∅).
      { rewrite H3; unfold Cartesian; apply AxiomII_P.
        repeat split; try apply Theorem49; Ens; apply AxiomII; Ens. }
      generalize (Theorem16 ([x,A])); intros; contradiction. }
  destruct H0; apply Ze.Zermelo in H0;
  try destruct H0 as [D H0], H0; Ens.
  intros; unfold En_p in H2; destruct H2; apply AxiomII in H2.
  apply AxiomII in H4;
  destruct H2, H4, H5 as [A H5], H6 as [B H6], H5, H6.
  rewrite H7, H8 in *; clear H7 H8; apply AxiomI; split; intros.
  - apply Theorem4' in H7; destruct H7; PP H7 a b; clear H7.
    apply AxiomII_P in H8; apply AxiomII_P in H9;
    destruct H8, H9, H8, H10.
    unfold Singleton in H11, H12; apply AxiomII in H11;
    apply AxiomII in H12.
    destruct H11 as [_ H11], H12 as [_ H12]; AssE A; AssE B.
    apply Theorem19 in H13; apply Theorem19 in H14; apply H12 in H13.
    apply H11 in H14. rewrite H13 in H14; rewrite H14 in H3;
    destruct H3; Ens.
  - generalize (Theorem16 z); intros; contradiction. }
destruct H0 as [D H0].
set (fc := \{\ λ A B, A ∈ (pow(X) ~ [∅]) /\
B = First( ∩((A×[A]) ∩ D)) \}\).
assert (Function (fc)).
{ unfold Function; split; intros.
  - unfold Relation; intros; PP H1 a b; Ens.
  - destruct H1; apply AxiomII_P in H1; apply AxiomII_P in H2.
    destruct H1, H2, H3, H4; rewrite H5, H6; auto. }
exists fc;unfold Choice_Function;repeat split;try apply H1;intros.
- clear H1; unfold Included, Range; intros; apply AxiomII in H1.
  destruct H1, H2; apply AxiomII_P in H2; clear H1; destruct H2, H2.
  apply Theorem49 in H1; destruct H1.
  assert ((x × [x]) ∈ (En_p X)).
  { unfold En_p; apply AxiomII; split; Ens.
    apply Theorem74; split; try apply Theorem42; auto. }
  AssE (x × [x]); apply H0 in H5; destruct H5; rewrite H5 in H3.
  assert (Ensemble x0).
  { apply Theorem42'; rewrite <- H5.
    apply Theorem33 with (x:=x×[x]);auto;unfold Included;intros.
    apply Theorem4' in H7; apply H7. }
  clear H6; double H7; apply Theorem44 in H7; destruct H7 as [H7 _].
  rewrite H7 in H3; clear H7.
  assert (x0 ∈ (x × [x] ∩ D)).
  { rewrite H5; unfold Singleton; apply AxiomII; Ens. }
  apply Theorem4' in H7; destruct H7 as [H7 _];PP H7 a b;clear H6 H7.
```

```
  apply AxiomII_P in H8; destruct H8, H7; apply Theorem49 in H6.
  apply Theorem54 in H6; destruct H6 as [H6 _]; rewrite H6 in H3.
  rewrite H3 in *; unfold Setminus, PowerSet in H2;
  apply Theorem4' in H2.
  destruct H2 as [H2 _]; apply AxiomII in H2; apply H2 in H7; auto.
- clear H1; apply AxiomI; split; intros.
  + unfold Domain in H1; apply AxiomII in H1; destruct H1, H2.
    apply AxiomII_P in H2; apply H2.
  + unfold Domain; apply AxiomII; split; Ens.
    assert ((z × [z]) ∈ (En_p X)).
    { unfold En_p; apply AxiomII; split.
      - apply Theorem74; split; try apply Theorem42; Ens.
      - exists z; split; auto. }
    AssE (z × [z]); apply H0 in H2; destruct H2.
    assert (Ensemble x).
    { apply Theorem42'; rewrite <- H2.
      apply Theorem33 with (x:= z × [z]); auto;
      unfold Included; intros.
      apply Theorem4' in H4; apply H4. }
    assert (x ∈ (z × [z] ∩ D)); clear H3.
    { rewrite H2; unfold Singleton; apply AxiomII; Ens. }
    apply Theorem4' in H5; destruct H5 as [H3 _]; PP H3 a b.
    clear H3; apply Theorem49 in H4; elim H4; intros; clear H6.
    apply Theorem54 in H4; destruct H4 as [H4 _];
    apply AxiomII_P in H5.
    destruct H5, H6; exists a; apply AxiomII_P; repeat split; auto.
    * apply Theorem49; split; Ens.
    * rewrite H2; apply Theorem44 in H5; destruct H5 as [H5 _].
      rewrite H5, H4; auto.
- apply Property_Value in H2; auto; apply AxiomII_P in H2;
  destruct H2, H3.
  assert ((A × [A]) ∈ (En_p X)).
  { unfold En_p; apply AxiomII; split.
    - apply Theorem74; split; try apply Theorem42; Ens.
    - exists A; split; auto. }
  AssE (A × [A]); apply H0 in H5; destruct H5.
  assert (Ensemble x).
  { apply Theorem42'; rewrite <- H5.
    apply Theorem33 with (x:= A × [A]); auto; unfold Included; intros.
    apply Theorem4' in H7; apply H7. }
  assert (x ∈ (A × [A] ∩ D)); clear H6.
  { rewrite H5; unfold Singleton; apply AxiomII; Ens. }
  apply Theorem4' in H8; destruct H8 as [H6 _]; PP H6 a b.
  clear H6; apply Theorem49 in H7; apply Theorem54 in H7.
  destruct H7 as [H7 _]; apply AxiomII_P in H8; destruct H8, H8.
```

```
    rewrite H5 in H4; apply Theorem44 in H6; destruct H6 as [H6 _].
    rewrite H6, H7 in H4; rewrite H4; auto.
Qed.

Hint Resolve Zermelo_AC : Axiom_of_Choice.

End Zermelo_Proof_AC.
```

4.10　Tukey 引理证明选择公理

本节将选择公理视为一个定理, 通过 Tukey 引理给出其证明, Coq 描述及代码如下:

```
Require Export Tukey_Lemma.

(* The Proof of Axiom of Choice *)

Module Type AC_Proof.

Declare Module Tu : Tukey_Lemma.

(* Special form of Choice Function *)

Definition Choice_Function' ε u : Prop :=
  Ensemble u /\ Function ε /\ dom(ε) = u /\ ran(ε) ⊂ ⋃u
  /\ (∀ x, x ∈ dom(ε) -> ε[x] ∈ x).

(* Hypotheses *)

Definition En_f X :=
  \{ λ g, ∃ A, A ⊂ (pow(X)~[∅]) /\ Choice_Function' g A \}.

(* Properties *)

Lemma Property_CF x : Ensemble x -> Choice_Function' ∅ ∅.
Proof.
  intros.
  unfold Choice_Function'; repeat split; intros.
  - generalize (Theorem26 x); intros.
    apply Theorem33 in H0; auto.
  - unfold Relation; intros.
    generalize (Theorem16 z); contradiction.
```

```
  - destruct H0; generalize (Theorem16 ([x0,y])).
    intros; contradiction.
  - apply AxiomI; split; intros.
    + unfold Domain in H0; apply AxiomII in H0; destruct H0, H1.
      generalize (Theorem16 ([z,x0])); contradiction.
    + generalize (Theorem16 z); contradiction.
  - unfold Included; intros; unfold Range in H0.
    apply AxiomII in H0; destruct H0, H1.
    generalize (Theorem16 ([x0,z])); contradiction.
  - unfold Domain in H0; apply AxiomII in H0; destruct H0, H1.
    generalize (Theorem16 ([x0,x1])); contradiction.
Qed.

Lemma Included_Function : ∀ (g f: Class),
  Function g /\ Function f /\ g ⊂ f -> (dom(g) ⊂ dom(f))
  /\ (∀ x, x ∈ dom(g) -> g[x] = f[x]).
Proof.
  intros.
  destruct H, H0.
  unfold Included in H1; split.
  - unfold Included; intros.
    unfold Domain in H2; apply AxiomII in H2; destruct H2, H3.
    unfold Domain; apply AxiomII; split; auto.
    exists x; apply H1 in H3; auto.
  - intros; apply Property_Value in H2; auto.
    apply H1 in H2; apply Theorem70 in H0; auto.
    rewrite H0 in H2; apply AxiomII_P in H2; apply H2.
Qed.

(* Lemma *)

Lemma Lemma_Fin_not_Em : ∀ X,
  Ensemble X -> FiniteSet (En_f X) /\ (En_f X) ≠ ∅.
Proof.
  intros.
  assert ((En_f X) ≠ ∅).
  { apply Property_NotEmpty; auto.
    exists ∅; unfold En_f; apply AxiomII; split.
    - assert (∅ ⊂ X); try apply Theorem26.
      apply Theorem33 in H0; auto.
    - exists ∅; split; try apply Theorem26.
      apply (Property_CF X); auto. }
  split; auto; unfold FiniteSet; repeat split; intros.
  - assert (En_f X ⊂ pow(pow(X) × (⋃pow(X)))).
    { unfold Included; intros.
```

```
    unfold En_f in H1; apply AxiomII in H1; destruct H1.
    destruct H2 as [A H2]; destruct H2.
    assert (pow(X) ∼ [∅] ⊂ pow(X)).
    { unfold Setminus, Included; intros.
      apply Theorem4' in H4; apply H4. }
    add (pow(X)∼[∅] ⊂ pow( X)) H2; apply Theorem28 in H2.
    unfold Choice_Function' in H3; destruct H3, H5, H6, H7.
    unfold PowerSet at 1; apply AxiomII; split.
    - rewrite <- H6 in H3; apply Theorem75; auto.
    - unfold Included; intros; unfold Cartesian; double H9.
      apply H5 in H10; destruct H10 as [a [b H10]]; rewrite H10 in *.
      clear H10; apply AxiomII_P; repeat split; Ens.
      + apply Property_dom in H9; rewrite H6 in H9.
        unfold Included in H2; apply H2; auto.
      + apply Property_ran in H9.
        unfold Included in H7; apply H7 in H9.
        apply Theorem31 in H2; destruct H2.
        unfold Included in H2; apply H2; auto. }
  apply Theorem38 in H; auto; double H.
  apply AxiomVI in H2; add (Ensemble (⋃ pow( X))) H.
  apply Theorem74 in H; apply Theorem38 in H; auto.
  apply Theorem33 in H1; auto.
- destruct H2; unfold En_f in H1; unfold En_f.
  apply AxiomII in H1; apply AxiomII; destruct H1.
  destruct H4 as [D H4], H4; unfold Choice_Function' in H5.
  destruct H5, H6, H7, H8; split.
  + apply Theorem33 in H2; auto.
  + generalize (classic (z=∅)); intros; destruct H10.
    * exists ∅; split; try apply Theorem26.
      rewrite H10; apply (Property_CF X); auto.
    * assert (Function z).
      { unfold Function in H6; destruct H6.
        unfold Function; split; intros.
        - unfold Relation in H6; unfold Relation; intros.
          unfold Included in H2; apply H2 in H12.
          apply H6 in H12; destruct H12, H12; exists x, x0; auto.
        - destruct H12; unfold Included in H2.
          apply H2 in H12; apply H2 in H13.
          add ([x,z0] ∈ F) H12; apply H11 in H12; auto. }
      double H11; add (Function F /\ z⊂F) H11.
      apply Included_Function in H11; destruct H11.
      exists (dom(z)); split.
      -- rewrite H7 in H11; add (D ⊂ pow(X)∼[∅]) H11.
         apply Theorem28 in H11; auto.
      -- { unfold Choice_Function';repeat split;try apply H12;auto.
```

```
          - rewrite H7 in H11; apply Theorem33 in H11; auto.
          - unfold Included; intros.
            unfold Element_U; apply AxiomII.
            unfold Range in H14; apply AxiomII in H14.
            destruct H14, H15; double H15; split; auto.
            apply Theorem70 in H12; auto.
            rewrite H12 in H15; apply AxiomII_P in H15.
            apply Property_dom in H16; destruct H15.
            exists x; split; auto; rewrite H17; double H16.
            unfold Included in H11; apply H11 in H16.
            apply H9 in H16; apply H13 in H18; rewrite H18; auto.
          - intros; double H14.
            unfold Included in H11; apply H11 in H14.
            apply H9 in H14; apply H13 in H15; rewrite H15; auto. }
- intros; destruct H1.
  unfold En_f; apply AxiomII; split; auto.
  assert (F ⊂ ⋃(En_f X) /\ ⋃(En_f X) ⊂ ((pow( X) ~ [∅]) × X)).
  { split.
    - unfold Included; intros; AssE z.
      unfold Element_U; apply AxiomII; split; auto.
      exists [z]; split.
      + unfold Singleton; apply AxiomII; split; auto.
      + apply H2; split; try apply Finite_Single; auto.
        unfold Included; intros; unfold Singleton in H5;
        apply AxiomII in H5.
        destruct H5; apply Theorem19 in H4; apply H6 in H4;
        rewrite H4; auto.
    - unfold Included; intros.
      unfold Element_U in H3; apply AxiomII in H3.
      destruct H3, H4 as [f1 H4], H4.
      assert (exists a b, z = [a,b]).
      { unfold En_f in H5; apply AxiomII in H5; destruct H5, H6, H6.
        unfold Choice_Function' in H7; apply H7 in H4; apply H4. }
      destruct H6 as [a [b H6]]; unfold Cartesian.
      rewrite H6; rewrite H6 in H3; rewrite H6 in H4.
      apply AxiomII_P; split; auto.
      unfold En_f in H5; apply AxiomII in H5.
      destruct H5, H7 as [A H7], H7.
      unfold Choice_Function' in H8.
      destruct H8, H9, H10, H11; split.
      + apply Property_dom in H4; rewrite H10 in H4.
        unfold Included in H7; apply H7 in H4; auto.
      + apply Property_ran in H4.
        unfold Included in H11; apply H11 in H4.
        unfold Element_U in H4; apply AxiomII in H4.
```

```
      destruct H4, H13, H13.
      unfold Included in H7; apply H7 in H14.
      unfold Setminus in H14; apply Theorem4' in H14; destruct H14.
      unfold PowerSet in H14; apply AxiomII in H14; destruct H14.
      unfold Included in H16; apply H16 in H13; auto. }
elim H3; intros; apply Theorem28 in H3.
generalize (classic (F = ∅)); intros; destruct H6.
+ exists ∅; split; try apply Theorem26.
  rewrite H6; apply (Property_CF X); auto.
+ assert (Function F).
  { unfold Function, Relation; split; intros.
    - unfold Included in H3; apply H3 in H7; PP H7 a b; Ens.
    - destruct H7.
      assert ([[x,y]|[x,z]] ⊂ F /\ Finite ([[x,y]|[x,z]])).
      { unfold Included, Unordered; split; intros.
        - apply AxiomII in H9; destruct H9, H10.
          + unfold Singleton in H10; apply AxiomII in H10.
            destruct H10; rewrite H11; auto; apply Theorem19; Ens.
          + unfold Singleton in H10; apply AxiomII in H10.
            destruct H10; rewrite H11; auto; apply Theorem19; Ens.
        - apply Theorem168; split; apply Finite_Single; Ens. }
      apply H2 in H9; unfold En_f in H9; apply AxiomII in H9.
      destruct H9, H10, H10; unfold Choice_Function' in H11.
      destruct H11, H12, H13; unfold Function in H12.
      apply H12 with (x0:=x); split; unfold Unordered; apply AxiomII.
      + split; try left; Ens; unfold Singleton; apply AxiomII; Ens.
      + split; try right; Ens; unfold Singleton;
        apply AxiomII; Ens.}
  exists dom(F); split.
  * unfold Included; intros; apply Property_Value in H8; auto.
    unfold Included in H3; apply H3 in H8.
    unfold Cartesian in H8; apply AxiomII_P in H8; apply H8.
  * { unfold Choice_Function'; repeat split;
      try apply H7; auto; intros.
      - assert (dom(F) ⊂ pow(X)).
        { unfold Included; intros.
          unfold Domain in H8; apply AxiomII in H8; destruct H8, H9.
          unfold Included in H3; apply H3 in H9.
          unfold Cartesian in H9; apply AxiomII_P in H9;
          destruct H9, H10.
          unfold Setminus in H10; apply Theorem4' in H10;
          apply H10.}
        apply Theorem33 in H8; apply Theorem38 in H; auto.
      - unfold Included; intros; apply AxiomII; split; Ens.
        unfold Range in H8; apply AxiomII in H8; destruct H8, H9.
```

```
          double H9; exists x; apply Property_dom in H9; split; auto.
          unfold Element_U in H4; apply H4 in H10.
          apply AxiomII in H10; destruct H10, H11 as [f H11], H11.
          unfold En_f in H12; apply AxiomII in H12;
          destruct H12, H13, H13.
          unfold Choice_Function' in H14; destruct H14, H15, H16, H17.
          double H11; apply Property_dom in H19; double H19.
          apply Property_Value in H20; auto; apply H18 in H19.
          add ([x,z] ∈ f) H20; unfold Function in H15.
          apply H15 in H20; rewrite H20 in H19; auto.
        - apply Property_Value in H8; auto; apply H4 in H8.
          unfold Element_U in H8; apply AxiomII in H8.
          destruct H8, H9 as [f H9], H9; unfold En_f in H10.
          apply AxiomII in H10; destruct H10, H11, H11.
          unfold Choice_Function' in H12; destruct H12, H13, H14, H15.
          double H9; apply Property_dom in H17; double H17.
          apply H16 in H17; apply Property_Value in H18; auto.
          add ([x,F[x]] ∈ f) H18; unfold Function in H13.
          apply H13 in H18; rewrite H18 in H17; auto. }
Qed.

Theorem Tukey_Choice : ∀ X,
  Ensemble X -> ∃ ε, Choice_Function ε X.
Proof.
  intros.
  double H; apply Lemma_Fin_not_Em in H0.
  apply Tu.Tukey in H0; destruct H0 as [F H0]; exists F.
  assert ((En_f X) ≠ ∅).
  { apply Property_NotEmpty; exists ∅; unfold En_f;
    apply AxiomII; split.
    - assert (∅⊂X);try apply Theorem26; apply Theorem33 in H1;auto.
    - exists ∅; split; try apply Theorem26;
      apply (Property_CF X); auto. }
  unfold MaxMember in H0; apply H0 in H1; clear H0.
  assert (Ensemble (En_f X)).
  { assert (En_f X ⊂ pow(pow(X) × (⋃pow(X)))).
    { clear H1; unfold Included; intros; unfold En_f in H0.
      apply AxiomII in H0; destruct H0, H1 as [A H1]; destruct H1.
      assert (pow(X) ~ [∅] ⊂ pow(X)).
      { unfold Setminus, Included; intros;
        apply Theorem4' in H3; apply H3. }
      add (pow(X)~[∅] ⊂ pow( X)) H1; apply Theorem28 in H1.
      unfold Choice_Function' in H2; destruct H2, H4, H5, H6.
      unfold PowerSet at 1; apply AxiomII; split.
      - rewrite <- H5 in H2; apply Theorem75; auto.
```

```
    - unfold Included; intros; unfold Cartesian; double H8.
      apply H4 in H9; destruct H9 as [a [b H9]]; rewrite H9 in *.
      clear H9; apply AxiomII_P; repeat split; Ens.
      + apply Property_dom in H8; rewrite H5 in H8; apply H1; auto.
      + apply Property_ran in H8; unfold Included in H6;
        apply H6 in H8.
        apply Theorem31 in H1; destruct H1; apply H1; auto. }
  apply Theorem38 in H; auto; double H; apply AxiomVI in H2.
  add (Ensemble (⋃ pow( X))) H; apply Theorem74 in H.
  apply Theorem38 in H; auto; apply Theorem33 in H0; auto. }
destruct H1; double H1; unfold En_f in H3; apply AxiomII in H3.
destruct H3, H4 as [D H4], H4; apply Property_ProperIncluded in H4;
destruct H4.
- double H4; apply Property_ProperIncluded' in H6;
  destruct H6 as [E H6], H6.
  double H6; unfold Setminus in H8;apply Theorem4' in H8;destruct H8.
  clear H8; unfold Complement in H9;apply AxiomII in H9;destruct H9.
  unfold NotIn in H9; assert (E ∈ [∅] <-> Ensemble E /\ (E=∅)).
  { split; intros.
    - unfold Singleton in H10; apply AxiomII in H10; destruct H10.
      split; auto; apply H11; apply Theorem19.
      generalize (Theorem26 E); intros; apply Theorem33 in H12; auto.
    - destruct H10; unfold Singleton; apply AxiomII; split; auto. }
  apply Lemma_z in H10; auto; clear H9.
  apply not_and_or in H10; destruct H10; try contradiction.
  apply Property_NotEmpty in H9; destruct H9 as [e H9].
  cut ((F ∪ [[E,e]])∈ (En_f X) /\ F ⊊ (F ∪ [[E,e]])); intros.
  { destruct H10; apply H2 in H10; contradiction. }
  assert (Ensemble ([E,e])). { apply Theorem49; split; Ens. }
  unfold Choice_Function' in H5; destruct H5, H11, H12, H13.
  assert (F ⊊ (F ∪ [[E,e]])).
  { unfold ProperIncluded; split.
    - unfold Included; intros; apply Theorem4; left; auto.
    - intro; rewrite <- H12 in H7; assert ([E,e]∈F).
      { rewrite H15; apply Theorem4; right.
        unfold Singleton; apply AxiomII; split; auto. }
      apply Property_dom in H16; contradiction. }
  split; auto; unfold En_f; apply AxiomII; split.
  + apply AxiomIV; split; auto; apply Theorem42; auto.
  + { exists (D ∪ [E]); split.
      - unfold Included; intros; apply Theorem4 in H16; destruct H16.
        + unfold ProperIncluded in H4; destruct H4.
          unfold Included in H4; apply H4 in H16; auto.
        + unfold Singleton in H16; apply AxiomII in H16.
          destruct H8; destruct H16; rewrite H17; auto.
```

```
      apply Theorem19; Ens.
    - assert (Function (F ∪ [[E,e]])).
      { unfold Function; split.
        - unfold Function in H11; destruct H11.
          unfold Relation in H11; unfold Relation; intros.
          apply Theorem4 in H17; destruct H17.
          + apply H11 in H17; auto.
          + unfold Singleton in H17; apply AxiomII in H17.
            exists E, e; apply H17; apply Theorem19; auto.
        - intros; destruct H16; apply Theorem4 in H16.
          apply Theorem4 in H17; destruct H16, H17.
          + unfold Function in H11; apply H11 with (x:=x); auto.
          + apply Property_dom in H16; rewrite H12 in H16.
            double H10;unfold Singleton in H17;apply AxiomII in H17.
            destruct H17; apply Theorem19 in H18; apply H19 in H18.
            apply Theorem49 in H10;apply (Theorem55 _ _ x z) in H10.
            symmetry in H18; apply H10 in H18; destruct H18.
            rewrite H18 in H7; contradiction.
          + apply Property_dom in H17; rewrite H12 in H17.
            double H10;unfold Singleton in H16;apply AxiomII in H16.
            destruct H16; apply Theorem19 in H18;apply H19 in H18.
            apply Theorem49 in H10;apply (Theorem55 _ _ x y) in H10.
            symmetry in H18; apply H10 in H18; destruct H18.
            rewrite H18 in H7; contradiction.
          + unfold Singleton in H16, H17.
            apply AxiomII in H16; apply AxiomII in H17.
            destruct H16, H17; double H10; apply Theorem19 in H20.
            double H20; apply H18 in H20; apply H19 in H21;
            double H10.
            apply Theorem49 in H10;apply (Theorem55 _ _ x y) in H10.
            apply Theorem49 in H22;apply (Theorem55 _ _ x z) in H22.
            symmetry in H20,H21;apply H10 in H20;apply H22 in H21.
            destruct H20, H21; rewrite <- H23, <- H24; auto. }
      assert (dom((F ∪ [[E,e]])) = D ∪ [E]).
      { apply AxiomI; split; intros.
        - unfold Domain in H17; apply AxiomII in H17.
          destruct H17, H18; apply Theorem4 in H18; destruct H18.
          + apply Property_dom in H18; rewrite H12 in H18.
            apply Theorem4; tauto.
          + unfold Singleton in H18;apply AxiomII in H18;double H10.
            destruct H18; apply Theorem19 in H10; apply H20 in H10.
            apply Theorem49 in H19;apply (Theorem55 _ _ z x)in H19.
            symmetry in H10; apply H19 in H10; destruct H10.
            apply Theorem4; right; unfold Singleton.
            apply AxiomII; split; auto.
```

```
    - unfold Domain; apply AxiomII; split; Ens.
      apply Theorem4 in H17; destruct H17.
      + exists F[z]; apply Theorem4; left.
        rewrite <- H12 in H17; apply Property_Value in H17; auto.
      + exists e; apply Theorem4; right; unfold Singleton in H17.
        apply AxiomII in H17; destruct H17.
        rewrite H18; try (apply Theorem19; Ens).
        unfold Singleton; apply AxiomII; split; auto. }
  unfold Choice_Function'; repeat split; try apply H16; auto.
  + apply AxiomIV; split; try apply Theorem42; Ens.
  + unfold Included; intros.
    unfold Range in H18; apply AxiomII in H18; destruct H18, H19.
    apply Theorem4 in H19; destruct H19.
    * unfold Element_U; apply AxiomII; split; auto.
      apply Property_ran in H19.
      unfold Included in H13; apply H13 in H19.
      unfold Element_U in H19; apply AxiomII in H19.
      destruct H19, H20, H20; exists x0; split; auto.
      apply Theorem4; tauto.
    * unfold Singleton in H10; apply AxiomII in H19.
      double H10; apply Theorem19 in H20; destruct H19.
      apply H21 in H20; apply Theorem49 in H10.
      apply (Theorem55 _ _ x z) in H10; symmetry in H20.
      apply H10 in H20; destruct H20; unfold Element_U.
      apply AxiomII; split; auto; exists E; rewrite <- H22.
      split; auto; apply Theorem4; right.
      unfold Singleton; apply AxiomII; split; Ens.
  + intros; unfold ProperIncluded in H15; destruct H15.
    add (Function (F ∪ [[E, e]]) /\ F ⊂ (F ∪ [[E, e]])) H11.
    apply Included_Function in H11; destruct H11.
    rewrite H17 in H18; apply Theorem4 in H18; destruct H18.
    * rewrite <- H12 in H18; double H18; apply H14 in H18.
      apply H20 in H21; rewrite <- H21; auto.
    * unfold Singleton in H18;apply AxiomII in H18;destruct H18.
      assert (Ensemble E); Ens.
      apply Theorem19 in H22; apply H21 in H22.
      assert (E ∈ dom(F ∪ [[E, e]])).
      { rewrite H17; apply Theorem4; right.
        unfold Singleton; apply AxiomII; auto. }
      apply Property_Value in H23; auto.
      assert ([E,e] ∈ (F ∪ [[E, e]])).
      { apply Theorem4; right.
        unfold Singleton; apply AxiomII; auto. }
      pattern E at 3 in H23; rewrite <- H22 in H23.
      add ([E, e] ∈ (F ∪ [[E,e]])) H23; unfold Function in H16.
```

```
          apply H16 in H23; rewrite H23; rewrite H22; auto. }
  - rewrite H4 in H5.
    unfold Choice_Function' in H5; unfold Choice_Function.
    destruct H5, H6, H7, H8; repeat split; try apply H6; auto.
    unfold Included; intros; unfold Included in H8.
    apply H8 in H10; unfold Element_U in H10.
    apply AxiomII in H10; destruct H10, H11, H11.
    unfold Setminus in H12; apply Theorem4' in H12.
    destruct H12; unfold PowerSet in H12.
    apply AxiomII in H12; destruct H12.
    unfold Included in H14; apply H14 in H11; auto.
Qed.

Hint Resolve Tukey_Choice : Axiom_of_Choice.

End AC_Proof.
```

Tukey 引理的一个深刻应用是一般拓扑学中著名的 Tychonoff 乘积定理[41, 84]的证明.

定理 (Tychonoff 乘积定理) *任何一族紧致空间的积空间都是紧致的.*

事实上, Tychonoff 乘积定理与选择公理也是等价的 [40].

我们当然可以继续对 Tychonoff 乘积定理及其与选择公理的等价性进行形式化证明实现 [88], 但这需要引进较多的拓扑学中的概念, 这里不准备展开了. 实现拓扑学的机器证明是吴文俊院士的一个宿愿. 本书作者团队已初步开展了 “点集拓扑学” 的形式化理论构建研究, 希望在不久的将来, 能较为完整地呈现给读者.

第 5 章　结论与注记

布尔巴基学派的序结构、代数结构、拓扑结构三大母结构是现代数学的基础. 利用计算机证明辅助工具, 可以完整地构建这三大母结构的形式化系统. 该系统可方便地应用于拓扑学和代数学理论的形式化构建.

本书利用交互式定理证明工具 Coq, 实现 Morse-Kelley 公理化集合论形式化系统, 整个形式化工作大约代码 10000 余行, 包括对该体系中 8 个公理 (含选择公理) 和 1 个公理图示以及全部 181 条定义或定理的 Coq 描述, 其中构造了序数和基数, 定义了非负整数, 把 Peano 公设当作定理, 可以迅速而自然地给出一个数学基础, 摆脱了明显的悖论. 进一步地, 作为系统的应用, 给出选择公理与它的几个著名等价命题间等价性的机器证明, 这些命题包括 Tukey 引理、Hausdorff 极大原则、极大原则、Zorn 引理、良序定理及 Zermelo 假定等. 在我们开发的系统中, 全部定理无例外地给出 Coq 的机器证明代码, 所有形式化过程已被 Coq8.9.0 验证, 并在计算机上运行通过. 需要说明的是, 在本书给出代码的过程中, 为了增加可读性, 将一些非纯文本表示的数学字符, 用一些标准的数学字符替换了. 我们的完整代码, 读者可通过扫描本书封底的二维码下载.

表 5.1 和表 5.2 分别提供了公理化集合论形式化系统和选择公理与其等价命题形式化相关文件的简要说明和统计数据. 表中标注了每个文件所对应的章节, 并从规范和证明两方面统计了每个文件的代码行数.

本形式化系统可方便地应用于拓扑学和代数学理论的形式化构建. 较之 MK 体系 [41], 本书除了完整实现其形式化外, 有些内容的改进是深入的和细致的, 例如为了逻辑上的严密性, 我们调整了一些概念的提出顺序, 增加了某些定理的必要条件, 删减了个别定理多余的条件, 此外还补充了一些常用的定义、引理和定理, 有些定理还给出了多种不同的机器证明过程, 系统性地给出了选择公理与几个著名命题等价性的机器证明.

表 5.1　公理化集合论形式化系统代码量统计

文件	对应章节	规范	证明
A_0.v (基本逻辑)	第 2 章	20	6
A_1.v (分类公理图示)	3.1 节	40	0
A_2.v (分类公理图示 (续))	3.2 节	10	0
A_3.v (类的初等代数)	3.3 节	240	260

续表

文件	对应章节	规范	证明
A_4.v (集的存在性)	3.4 节	120	200
A_5.v (序偶：关系)	3.5 节	100	260
A_6.v (函数)	3.6 节	120	240
A_7.v (良序)	3.7 节	150	750
A_8.v (序数)	3.8 节	160	760
A_9.v (非负整数)	3.9 节	80	300
A_10.v (选择公理)	3.10 节	60	320
A_11.v (基数)	3.11 节	780	3000

表 5.2　选择公理及其等价命题形式化代码量统计

文件	对应章节	规范	证明
Basic_Definitions.v	4.1 节	150	110
Tukey_Lemma.v	4.2 节	90	340
Hausdorff_Maximal_Principle.v	4.3 节	30	120
Maximal_Principle.v	4.4 节	20	50
Zermelo_Postulate.v	4.5 节	20	280
Zorn_Lemma.v	4.6 节	30	240
Wellordering_Theorem.v	4.7 节	170	430
WO_Proof_AC.v	4.8 节	10	80
Zermelo_Proof_AC.v	4.9 节	15	120
Tukey_Proof_AC.v	4.10 节	50	350

几点注记如下:

(1) Coq 定理的描述接近自然数学语言, 可以看成是一种 “翻译”, 当然, 这种 “翻译” 是建立在严格、准确的形式化基础上的. Coq 定理的证明过程具有规范、严格和可靠的特点. Coq 中所有的证明程序都严格按照已给出的定义、公理及已完成证明的定理形式化地进行.

(2) Coq 证明具有智能性的特点, 运行快速, 相比人工证明, Coq 证明省去了许多繁琐的推理时间, 我们可以直接在 Coq 中运行相应指令得到想要看到的结果.

(3) Coq 证明体现交互性, 如图 5.1 所示, 证明过程中我们在 Coq 软件平台的左侧编写证明, 在右侧上方实时地显示证明条件和证明目标, 充分体现 Coq 证明的可读性. 右侧下方实时显示证明是否成功, 若失败还能显示具体错误, 具有纠错能力.

本书中给出的代码是完整的. 在完成代码的过程中, 我们可以很自然地发现 Kelley 在《一般拓扑学》中的一些忽略的条件及打印错误, 第 3 章定理 61、定理 160 和定理 180 的证明即是明显的例子. 读者在计算机运行我们代码的过程中完全可以中断命令、逐条验证定理证明过程的每一细节 (当然可以对照人工证明), 包括人

工证明省略的部分, 从而既学习理解相关数学理论, 又实践提高计算机 Coq 技术, 将人的智慧与机器的智能结合起来, 充分体会基于 Coq 的数学定理机器证明的可读性和交互性特点, 以及证明过程的规范性、严谨性和可靠性. 本书相关工作已获国家计算机软件著作权登记证书 (图 5.2).

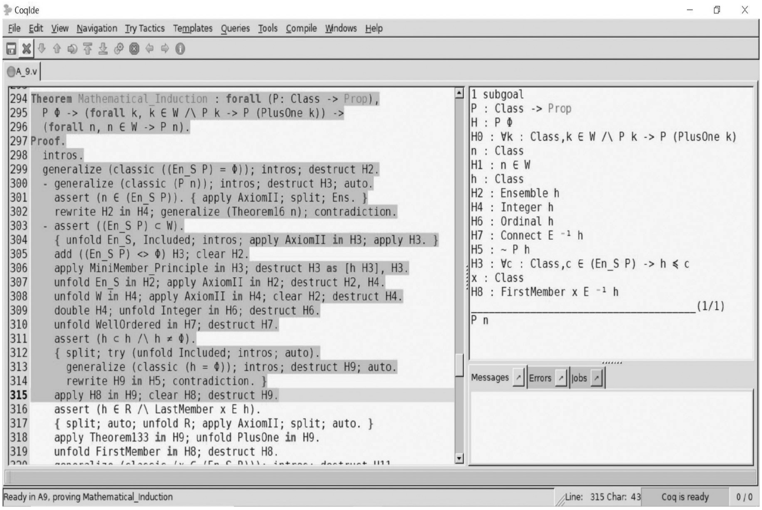

图 5.1　数学归纳法原理的证明截图 (后附彩图)

图 5.2　国家计算机软件著作权登记证书 (后附彩图)

本书是 Morse-Kelley 公理化集合论系统的首次形式化实现, 在此基础上, 现代数学的拓扑学和代数学理论可方便、快速地形式化构建. 本书也是第一本全部定理证明都由 Coq 代码完成的数学书, 在一定意义上, 实现了读者跟随计算机学习、理解、构建乃至教育现代数学的尝试.

在当今各学科都大力标举 “人工智能” 的背景下, 我们当然希望也能为人工智能的基础研究做点贡献, 希望本书如鲁迅先生在《坟•题记》中所说的: “但愿这本书能够暂时躺在书摊上的书堆里, 正如博厚的大地, 不至于容不下一点小土块.”①

最后, 需要说明的是, 本书中各定理给出的证明策略可能不是最优的, 证明代码也有进一步简化的余地, 读者完全可以根据自己的证明思路, 在我们代码的基础上, 充分利用 Coq 的智能性, 采用更为优化的策略, 写出更加可读、简明、甚至具有结构化[4] 的证明代码. 这当然要求读者既有对于数学思想的深刻理解, 又有对于 Coq 的熟练使用技巧, 这也是一种艺术, 需要长期实践积累, 也是今后努力的方向.

顺便提一下, 关于形式化数学, 在 20 世纪初对于数学基础的深入讨论中受到重视, 后来的法国布尔巴基学派对振兴法国当代数学起到了积极的推动作用, 也对整个 20 世纪数学的发展产生了深远的影响, 虽然有时也饱受过于 “形式” 的诟病 [39, 42, 59, 69]. 20 世纪 90 年代、特别是进入 21 世纪以来, 随着计算机形式化工具的出现, 尤其是 “四色定理”[24] ②、“有限单群分类定理”[25, 26] 及 “Kepler 猜想” [32] 等一系列著名数学难题形式化证明的实现, 使得证明辅助工具 Coq 在学术界得到广泛认可. 2002 年菲尔兹奖获得者 Voevodsky 和 2010 年菲尔兹奖获得者 Villani 都大力倡导发展可信数学 [72, 73], 他们指出, 当今数学论证变得如此复杂, 而计算机软件能够检查卷帙浩繁的数学证明正确性, 人类的大脑无法跟上数学不断增长的复杂性, 计算机检验将是唯一的解决方案 [72, 73]. 今后, 每一本严谨的数学专著, 甚至每一篇数学论文, 都可由计算机检验其细节的正确性, 这正发展为一种趋势. 英国帝国理工学院的纯数学教授 Buzzard 在剑桥举办的一次研讨会上表示: 证明是一种很高的标准, 我们不需要数学家像机器一样工作, 而是可以要他们去使用机器③!

形式化数学与计算机工具的结合, 前景大有可为! 让我们在数学与信息科学的交叉领域中充分感受类似西方哥特式建筑所具有的崇高、庄严、神圣、清峻和终极的美吧!

---

① 《鲁迅全集》第一卷: 4. 北京: 人民文学出版社, 2005.

② 关于 “四色定理”, 20 世纪 70 年代曾有一个基于计算机的证明 (证明中的关键部分是靠计算机算法验证的)[1], 目前的形式化证明 [24] 得到了数学界和计算机界的普遍认可, 它的每一个细节理论上都是可以论证清楚的. “四色定理” 的人工证明, 尚无确切的定论. 北京大学许进教授在一个会议上公布了一个人工证明, 并在多所大学报告了他的结果, 张景中院士曾建议用 Coq 验证许进教授的证明, 尚未看到最终的报道.

③ 科学: 纯数学陷入了危机? 见: http://mini.eastday.com/a/190404055449350.html

参 考 文 献

[1] Appel A, Haken W. Every planar map is four colourable. Bulletin of the American Mathematical Society, 1976, 82(5): 711-712.

[2] Appel A W, Dockins R, Hobor A, et al. Program Logics for Certified Compilers. New York: Cambridge University Press, 2014.

[3] Barras B. Sets in Coq, Coq in sets. Journal of Formalized Reasoning, 2010, 3(1): 29-48.

[4] Beeson M. Mixing computations and proofs. Journal of Formalized Reasoning, 2016, 9(1): 71-99.

[5] Belinfante J G F. On computer-assisted proofs in ordinal number theory. Journal of Automated Reasoning, 1996, 22(3): 341-378.

[6] Bell J L. Set Theory: Boolean-valued Models and Independence Proofs (Oxford Logic Guides 47). 3rd ed. Oxford: Clarendon Press, 2005.

[7] Bernays P, Fraenkel A A. Axiomatic Set Theory. Amsterdam: North Holland Publishing Company, 1958.

[8] Bertot Y, Castéran P. Interactive Theorem Proving and Program Development – Coq’Art: The Calculus of Inductive Constructions. Berlin: Spring-Verlag, 2004. (中译本, 交互式定理证明与程序开发——Coq 归纳构造演算的艺术. 顾明, 等译. 北京: 清华大学出版社, 2010.)

[9] Booker A R. Turing and the Riemann hypothesis. Notice of the American Mathematical Society, 2006, 53(10): 1208-1211.

[10] Bourbaki N. The architecture of mathematics. American Mathematical Monthly, 1950, 57(4): 221-232.

[11] Bourbaki N. Elements of Mathematics: Algebra I. Berlin: Spring-Verlag, 1989. Translation of the 1970 French edition.

[12] Bourbaki N. Elements of Mathematics: General Topology, Part 1. Berlin: Spring-Verlag, 1995. Translation of the 1966 French edition.

[13] Bourbaki N. Elements of Mathematics: Theory of Sets. Berlin: Spring-Verlag, 2004. Translation of the 1970 French edition.

[14] Cantor G. Contributions to the Founding of the Theory of Transfinite Numbers (Translated by Jourdain P E B). New York: Dover Publications, 1915.

[15] Chlipala A. Certified Programming with Dependent Types. Massachusetts: MIT Press, 2013.

[16] Cohen P J. Set Theory and the Continuum Hypothesis. New York: W A Benjamin. INC, 1966.

[17] The Coq Development Team. The Coq Proof Assistant Reference Manual (Version 8.9.0). https://coq.inria.fr/distrib/current/refman, 2019.

[18] Devlin K J. The Joy of Sets: Fundamentals of Contemporary Set Theory. 2nd ed. New York: Springer-Verlag, 1992.

[19] Fraenkel A A. Abstract Set Theory (Third revised edition). Amsterdam: North Holland Publishing Company, 1966.

[20] Fraenkel A A, Bar-Hillel Y, Levy A. Foundations of Set Theory (Second revised edition). Amsterdam: Elsevier, 1973.

[21] Gödel K. Consistency-proof for the generalized continuum-hypothesis. Proceedings of the National Academy of Sciences, 1939, 25(4): 220-224.

[22] Gödel K. The Consistency of the Axiom of Choice and of the Generalized Continuum Hypothesis With the Axioms of Set Theory. Princeton: Princeton University Press, 1940.

[23] Godement R. Algebra. Paris: Hermann, 1968.

[24] Gonthier G. Formal proof - the Four Color Theorem. Notices of the American Mathematical Society, 2008, 55(11): 1382-1393.

[25] Gonthier G. Feit thomson proved in coq. http://www.msr-inria.fr/news/feit-thomson-proved-in-coq/, 2012.

[26] Gonthier G, Asperti A, Avigad J, et al. Machine-checked proof of the Odd Order Theorem. Proceedings of the Interactive Theorem Proving 2013 (Blazy S, Paulin-Mohring C, Picharidie D, Eds), LNCS, 2013, 7998: 163-179.

[27] Grimm J. Implementation of Bourbaki's mathematics in Coq: part two, ordered sets, cardinals, integers. Research Report RR-7150, INRIA, 2009. http://hal.inria.fr/inria-00440786/en/.

[28] Grimm J. Implementation of Bourbaki's mathematics in Coq: part one, theory of sets. Journal of Formalized Reasoning, 2010, 3(1): 79-126.

[29] Grimm J. Implementation of Bourbaki's mathematics in Coq: part two, from natural to real numbers. Journal of Formalized Reasoning, 2016, 9(2): 1-52.

[30] Hales T C. Formal proof. Notices of the American Mathematical Society, 2008, 55(11): 1370-1380.

[31] Halmos P R. Naive Set Theory. New York: Springer-Verlag, 1974.

[32] Hales T, Adams M, Bauer G, et al. A formal proof of the Kepler conjecture. http://arxiv.org/pdf/1501.02155.pdf, 2015.

[33] Harrision J. Formal proof - theory and practice. Notices of the American Mathematical Society, 2008, 55(11): 1395-1406.

[34] Heijenoort J V. From Frege To Gödel: A Source Book in Mathematical Logic, 1879-1931. Cambridge: Harvard University Press, 1967.

[35] Huet G, Kahn G, Paulin-Mohring C. The Coq Proof Assistant: A Tutorial (Version 8.5). Technical Report 178, INRIA 2016. https://coq.inria.fr/tutorial/

[36] Huntington E V, Cantor G. The Continuum, and Other Types of Serial Order: With an Introduction to Cantor's Transfinite Numbers. 2nd ed. New York: Dover Publications, 2003.

[37] Jech T. The Axiom of Choice. Amsterdam: North Holland Publishing Company, 1973.

[38] Jech T. Set Theory (The third millennium edition, revised and expanded). Berlin: Springer-Verlag, 2003.

[39] Katz V J. A History of Mathematics: An Introduction. 3rd ed. Boston: Addison-Wesley, 2009.

[40] Kelley J L. The Tychonoff product theorem implies axiom of choice. Fund. Math., 1950, 37(1): 75-76.

[41] Kelley J L. General Topology. New York: Springer-Verlag, 1955. (中译本, 一般拓扑学. 吴从炘, 吴让泉译. 北京: 科学出版社, 2010.)

[42] Kline M. Mathematical Thought from Ancient to Moderns Times. Oxford: Oxford University Press, 1972.

[43] Kirst D, Smolka G. Categoricity results for second-order ZF in dependent type theory. Proceedings of the Interactive Theorem Proving 2013 (Ayala-Rincón M, Munoz C A, Ed.). LNCS, 2017, 10499: 304-318.

[44] Kirst D, Smolka G. Large model constructions for second-order ZF in dependent type theory. Andronick J, Felty A P, ed. Proceedings of CPP. ACM, 2018: 228-239.

[45] Landau E. Foundations of Analysis: The Arithmetic of Whole, Rational, Irrational, and Complex Numbers (Third edition, Translated by Steinhardt F). New York: Chelsea Publishing Company, 1966. Translation of the 1929 French edition.

[46] 陆汝钤. 人工智能. 北京: 科学出版社, 1988.

[47] 陆汝钤. 计算机系统的形式语义. 北京: 清华大学出版社, 2017.

[48] Mendelson E. Introduction to Mathematical Logic. 4th ed. London: Chapman and Hall, 1997.

[49] Monk J D. Introduction to Set Theory. New York: McGraw-Hill, 1969.

[50] Morse A P. A Theory of Sets. New York: Academic Press, 1965.

[51] Nipow T, Paulson L C, Wenzel M. Isabelle/HOL: A Proof Assistant for Higher-Order Logic. Berlin: Springer-Verlag, 2002.

[52] Paulson L C. Mechanizing set theory: Cardinal arithmetic and the Axiom of Choice. Journal of Automated Reasoning, 1996, 17(3): 291-323.

[53] Paulson L C. The relative consistency of the Axiom of Choice mechanized using Isabelle/ZF. LMS J. Comput. Math., 2003, (6): 198-248.

[54] Pierce B C, de Amorim A A, Casinghino C, et al. Software Foundation. http://softwarefoundations.cis.upenn.edu/, 2017.

[55] 戚征. 选择公理与连续统假设. 数学进展, 1984, 13(1): 4-22.

[56] Rubin H, Rubin J E. Equivalents of the Axiom of Choice. Amsterdam: North Holland Publishing Company, 1963.

[57] Rubin H, Rubin J E. Equivalents of the Axiom of Choice, II. Amsterdam: North Holland Publishing Company, 1985.

[58] Rubin J E. Set Theory for the Mathematician. San Francisco: Holden Day, 1967.

[59] Ruelle D. The Mathematician's Brain. Princeton and Oxford: Princeton University Press, 2007.

[60] Shen A, Vereshchagin N K. Basic Set Theory (Student Mathematical Library, Volume 17). American Mathematical Society, 2002.

[61] Shu R D, Yu W S. Mathematical theorem machine proving system based on Coq - Machine proving of the factorization theorem of principle ideal ring. Proceedings of 2017 China Intelligent Network of Things System Conference, 2017.

[62] 数学辞海总编辑委员会. 数学辞海. 太原：山西教育出版社; 南京：东南大学出版社; 北京: 中国科学技术出版社, 2002.

[63] Sun T Y, Yu W S. Machine proving system for mathematical theorems based on Coq - Machine proving of Hausdorff Maximal Principle and Zermelo Postulate. Proceedings of the 36th Chinese Control Conference, 2017: 9871-9878.

[64] 孙天宇, 郁文生. 基于 Coq 的选择公理及其等价命题的机器实现. 2017 中国智能物联系统会议, 2017.

[65] Sun T Y, Yu W S. Formalization of the Axiom of Choice and its equivalent

theorems. 2018 (https://arxiv.org/abs/1906.03930).
[66] Sun T Y, Yu W S. Formalization of axiomatic set theory in Coq. IEEE Access, 2020, 8: 21510-21523.
[67] 孙天宇, 郁文生. 选择公理与 Tukey 引理间等价性的形式化证明. 北京邮电大学学报 (自然科学版), 2019, 42(5): 1-7.
[68] Takeuti G, Zaring W M. Axiomatic Set Theory. New York: Springer-Verlag, 1973.
[69] Thurston W P. On proof and progress in mathematics. Bulletin (New Series) of the American Mathematical Society, 1994, 30(2): 161-177.
[70] Tukey J W. Convergence and Uniformity in Topology (Annals of Mathematics Studies 2). Princeton: Princeton University Press, 1940.
[71] Vaught R L. Set Theory: An Introduction. 2nd ed. Boston: Birkhäuser, 2001.
[72] Vivant C. Thèoréme Vivamt. Prais: Bernard Grasset, 2011. (中译本, 一个定理的诞生. 马跃, 杨苑艺译. 北京: 人民邮电出版社, 2016.)
[73] Voevodsky V. Univalent foundations of mathematics. Proceedings of the 18th international workshop on logic, language, information and computation. (Beklemishev L, De Queiroz R ed. WoLLIC 2011, Philadelphia, PA, USA) LNAI 2011, 6642: 4.
[74] 汪芳庭. 公理集论. 合肥: 中国科学技术大学出版社, 1995.
[75] 汪芳庭. 数学基础. 北京: 科学出版社, 2001.
[76] Wang H. On Zermelo's and Von Neumann's axioms for set theory. Proc. Natl. Acad. Sci., 1949, 35(3): 150-155.
[77] 文兰. 悖论的消解. 北京: 科学出版社, 2018.
[78] Werner B. Sets in types, types in sets. Proceedings of TACS, 1997: 530–546.
[79] Wiedijk F. Formal proof - getting started. Notices of the American Mathematical Society, 2008, 55(11): 1408-1414.
[80] Wu W T. Mechanical Theorem Proving in Geometries: Basic Principles (Translated by Jin X F, Wang D M). New York: Springer-Verlag, 1994. Translation of the 1984 Chinese edition.
[81] Wu W T. Mathematics Mechanization: Mechanical Geometry Theorem-Proving, Mechanical Geometry Problem-Solving, and Polynomial Equations-Solving // Hazewinkel M, ed. Mathematics and Its Applications, Volume 489, Beijing and Dordrecht: Science Press and Kluwer Academic Publishers, 2000.
[82] 萧文灿. 集合论初步. 北京: 商务印书馆, 1950.
[83] 谢邦杰. 超穷数与超穷论法. 长春: 吉林人民出版社, 1979.

[84] 熊金城. 点集拓扑讲义. 3 版. 北京: 高等教育出版社, 2003.

[85] Yang L, Xia B C. Automated Inequality Proving and Discovering. New Jersey: World Scientific Publishing Company, 2016.

[86] Yu Y. Computer proofs in group theory. Journal of Automated Reasoning, 1990, 6(3): 251-286.

[87] Yuan J, Yu W S. Formalization of modern algebra theory in Coq - Formal proof of the Rank-nulity theorem. Proceedings of 2018 China Intelligent Network of Things System Conference, 2018.

[88] Zao X Y, Sun T Y, Fu Y S, et al. Formalization of general topology in Coq - A formal proof of Tychonoff's theorem. Proceedings of the 38th Chinese Control Conference, 2019: 2685-2691.

[89] Zermelo E. Collected Works (Volume I: Set Theory) (von Herausgegeben, Ebbinghaus H D, Kanamori A, Ed). Berlin: Springer-Verlag, 2010.

[90] 张德学. 一般拓扑学基础. 北京: 科学出版社, 2012.

[91] 张禾瑞, 赫鋼新. 高等代数. 5 版. 北京: 高等教育出版社, 2007.

[92] 张锦文. 公理化集合论导论. 北京: 科学出版社, 1991.

[93] 张景中, 李永彬. 几何定理机器证明三十年. 系统科学与数学, 2009, 29(9): 1155-1168.

[94] 郑维行, 王声望. 实变函数与泛函分析概要 (第一册). 4 版. 北京: 高等教育出版社, 2010.

[95] Zorn M. A remark on method in transfinite algebra. Bulletin of the American Mathematical Society, 1935, 41: 1667-670.

索　　引

J

K

L

M

N

P

Q

R

S

T

W

彩　　图

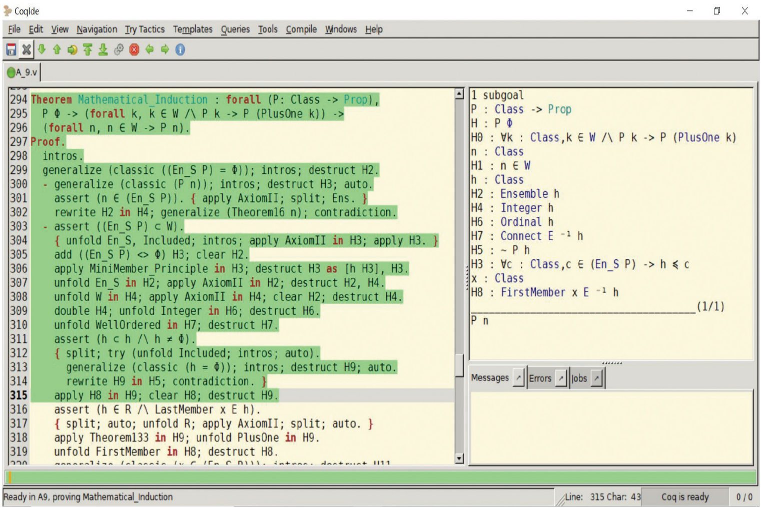

图 5.1　数学归纳法原理的证明截图

图 5.2　国家计算机软件著作权登记证书

(O-7955.31)

数学机械化丛书

1 数学机械化	2003.3	吴文俊 著
2 消去法及其应用	2002.8	王东明 著
3 实域论	2003.12	曾广兴 著
4 多项式系统的实根分离算法及其应用	2004.5	陆征一 等 著
5 近世计算理论导引	2004.6	黄文奇 等 著
6 可积系统与计算机代数	2004.6	范恩贵 著
7 方程求解与机器证明	2006.9	高小山 等 著
8 复杂非线性波的构造性理论及其应用	2007.3	闫振亚 著
9 交互式马尔可夫链	2007.6	吴尽昭 等 著
10 进程代数	2007.6	王永祥 等 著
11 不等式机器证明与自动发现	2008.1	杨 路 等 著
12 几何定理机器证明的几何不变量方法	2015.4	张景中 等 著
13 公理化集合论机器证明系统	2020.3	郁文生 等 著

www.sciencep.com

ISBN 978-7-03-064039-0

9 787030 640390 >

定 价：128.00 元

科学出版社互联网入口
科学数理分社
电 话：(010) 64010624
E-mail：wangliping@mail.sciencep.com

本书在线
资源获取

销售分类建议：定理机器证明 形式化数学